QL Today

Volume 13
Issue 2
Dec.-February
2008-09

ISSN 1432-5454

The Magazine about QL, QDOS, Sinclair Computers, SMSQ...

Do you remember the Visitors of QL 2004?

Are we going to have more of these Events in the Future? What about the Future of Eindhoven QL Meetings? More in this Issue!

www.QLToday.com

Contents

The deadline for the next issue is the 16th of February 2009

Advertisers

in alphabetical order

QL Today

ISSN 1432-5454

German office & Publisher:

Jochen Merz Software	Tel. +49 203 502011
Kaiser-Wilhelm-Str. 302	Fax +49 203 502012
47169 Duisburg	email: smsq@j-m-s.com
Germany	email: QLToday@j-m-s.com

English office:

Q Branch	Tel. +44 1273 430501
20 Locks Hill	Mobile +44 7836 745501
Portslade	Fax +44 1273 430501
BN41 2LB	email: qbranch@qbranch.demon.co.uk
United Kingdom	email: QLToday@j-m-s.com

Editor:

Geoff Wicks	Tel. +44 1332 271366
5b Wordsworth Avenue	email: gtwicks@btinternet.com
Sinfin	email: QLToday@j-m-s.com
Derby DE24 9HQ	
United Kingdom	

Co-Editor:

Bruce Nicholls	Tel +44 20 71930539
38 Derham Gardens	Fax +44 870 0568755
Upminster	email: qltoday@q-v-d.demon.co.uk
Essex RM14 3HA	email: QLToday@j-m-s.com
United Kingdom	

QL Today is published four times a year, our volume begins on beginning of June. Please contact the German or English office for current subscription rates or visit our homepage www.QLTODAY.com.

We welcome your comments, suggestions and articles. YOU make **QL Today** possible. We are constantly changing and adjusting to meet your needs and requirements. Articles for publication should be on a 3.5" disk (DD or HD) or sent via Email. We prefer ASCII, Quill or text87 format. Pictures may be in _SCR format, we can also handle GIF or TIF or JPG. To enhance your article you may wish to include Saved Screen dumps. PLEASE send a hardcopy of all screens to be included. Don't forget to specify where in the text you would like the screen placed.

QL Today reserves the right to publish or not publish any material submitted. Under no circumstances will **QL Today** be held liable for any direct, indirect or consequential damage or loss arising out of the use and/or inability to use any of the material published in **QL Today**. The opinions expressed herein are those of the authors and are not necessarily those of the publisher.

If you need more information about the UNZIP program which is used by our BOOT program to unpack the files, we suggest that you visit Dilwyn Jones' web site where you find more information about lots of interesting QDOS software and INFOZIP at **http://www.dilwyn.uk6.net/arch/index.html**

Editorial

by Geoff Wicks

This has not been a good autumn for QL publications. Both QL Today and the Quanta Magazine have had serious problems.

About half of QL Today readers received the last issue almost 6 weeks late because of problems at the UK side of the operation. The details are in our news section.

At Quanta a newly appointed editor appears to have gone AWOL and an, already over-worked, John Gilpin has had to resume the temporary editorship. This result is severe delays in publication and an emergency publishing schedule.

2008 has been a year of QL gloom. We have seen a drastic reduction in UK trading activity. For the first time ever there has been no Quanta show in the South of England. In this issue we bring news of changes in the Netherlands that could lead to the closure of Sin_QL_Air.

In the Spring Quanta had good news for its members, but even this has gone sour. On the positive side Dilwyn Jones is proving his worth as a new committee member and, in the background, work is slowly progressing on a reorganisation of the library. Nevertheless Quanta has recently been dogged by sheer bad luck that has put some progress on hold, and on some issues the officers have shown a lack of flexibility.

Amidst all the doom and gloom it is important to remember the QL will shortly celebrate its quarter centenary. Who would have predicted more than two decades ago, when most of the experts had condemned the QL to an early death, that 25 years on we would still be alive? How many of the PC's hardware, software and dot.com companies will be able to make this claim 25 years from now? Who could have predicted when Quanta was founded in 1984 that a quarter of a century on it would still have about 190 members? QL-ers have achieved something remarkable..

The QL survived through the hard work of many individuals. We have space to name just two who have helped to keep QL publications alive.

The first is John Gilpin. He has never officially been the editor of the Quanta Magazine, but always acting editor. In his early editorship he had to quickly learn to produce the magazine in Page Plus, and then guide it towards electronic publication. During his period of office the magazine has improved in reliability and content. Without John's contribution Quanta could have ceased to function about 3 years ago.

The second person may surprise some readers given recent events, but it is Roy Wood. There is a large part of the QL Today operation that remains invisible to the readers. It is a complex and time consuming task to produce the magazine, and over the years there have been many occasions when consideration was given to closing down. At these times it was Roy's energy and enthusiasm that kept others going. Without Roy QL Today could have disappeared some years ago.

The QL still needs workers like these.

QL Today at the Crossroads

The decline in QL trading activity in the UK has had a direct impact on QL Today with some readers getting the last issue 6 weeks late.

Earlier this year we reported on the decision of Tony Firshman to reduce his QL activities because of pressures on his time, and at QL Today we feared there would be a knock-on effect that would eventually affect the magazine. In particular we were aware that in recent years the trading pattern of QBranch had changed greatly in that the software market is now almost non-existent and what trading there is is largely in second hand hardware. We were also aware of increasing demands on Roy Wood's time from outside the QL community. Realistically we had to be prepared for the possibility of QBranch, through whom about half our readers subscribe, closing down. It was clear to us that the structure of the magazine in which the continental and UK arms have separate administration, distribution and financing was inappropriate for a situation in which UK trading activity was rapidly diminishing.

QL Today was founded in 1996 following the collapse of IQLR and the two founding fathers were Jochen Merz and Stuart Honeyball. Shortly after the magazine was founded Stuart ceased trading and QBranch took over as the UK office. Production of the magazine took place in Germany, but by far the largest group of subscribers were in the UK. It was therefore logical to set up two separate arms for the magazine, one on the continent and the other in the UK.

Technically those readers who subscribe to the magazine via the UK office do not have a direct subscription to QL Today. Their contact is with QBranch who then buy in the copies from the publisher. This structure has worked reasonably well over the years, but has come under increasing strain in the last 12 months. Matters came to a head in September of this year when Jochen Merz reported to the QL users email group:

"As I have not had an order from QBranch for their QL Today issues and no replies to emails, I cannot predict when the readers who subscribed with QBranch will get their issues."

As QL Today does not have names and addresses of readers who subscribe via QBranch we were unable to inform them directly.

A fortnight after Jochen's email QBranch had still not provided QL Today with the information required and QL Today's editor posted a second email on the QL Users email group in somewhat stronger terms. This produced a reaction from QBranch:

"This is completely my fault - although Geoff has over-reacted as usual. I have had little time to read this list of late since, apart from my family, my job, my current position as Diving Officer and instructor for Sussex Diving Club and my rock band, I seem to have got involved in a political campaign to stop a big building development in Brighton Marina.

I was supposed to furnish Jochen with the number of magazines I need for this current issue and I neglected to do so. Since then I have been so tied up in all these activities it has been push onto back brain. I apologise to those who subscribed for this oversight and I will rectify matters this week."

The editorial team of QL Today views the situation with considerable dismay. Following the move to quarterly publication last year we have been able to build up a rhythm in the production of the magazine that has enabled us to keep to our promised publication schedules. The last issue was published on time and those readers who subscribe via Jochen Merz received their copies promptly.

In a time when QL trading activity is rapidly declining it is essential that the magazine retains a reputation for reliability. Although QL Today remains editorially viable, the editorial team are becoming increasingly concerned that the number of regular writers is now at the very minimum to ensure the magazine's medium and long term future.

Over the years the readership pattern has changed and there are now roughly equal numbers of continental and UK subscribers. The changing readership pattern and reduction of UK trading activity suggest that the present structure of the magazine is no longer appropriate. QBranch expressed a wish to distribute the current volume, but QL Today will no longer make use of the UK office after that. All readers will then have a direct subscription to QL Today.

During his association with QL Today Roy Wood has had a distinguished record of service to the magazine. Production of the magazine over 13 years has not always been easy and during the difficult periods he had an infectious enthusiasm that gave others the encouragement to continue. The editorial team believes that without the contribution of Roy the magazine could have disappeared many years ago.

Website Review

During the last few weeks there have been a number of changes to QL websites, some minor, some more important and some that raise interesting policy questions. QL Today begins this special news feature on websites by looking at Quanta subgroups.

The 2004 demographic survey of Quanta membership showed that Quanta membership was not distributed evenly over the UK but that there were clusters of members. The areas in which there was a cluster were usually where there was or still is an active Quanta subgroup. At the moment Quanta has 8 subgroups. The most active are London, Manchester, Birmingham and Scotland. Two subgroups, Surrey and Sussex, were mainly known through the regular shows at Byfleet and Portslade but it is now uncertain how active they are. The remaining two are Dorset and Solent, but nothing is known of these, which raises the question whether they still exist in practice.

Quanta's webmaster, Dan Abott, has stated on several occasions that he would like to see the subgroups having an active role on Quanta's main website but there has been little response from the groups themselves. Two subgroups SQLUG (Scotland) and London have their own website and both have reported changes in the last few weeks.

SQLUG

The main function of the SQLUG website (**http://www.jms1.supanet.com**) is for the downloading of software by George Gwilt and others. George has now reported a bit of bad news plus a solution: *"The SQLUG website cannot now be updated. In order that programs for which I am responsible can be made available I have put some of the latest updates on the site at*
http://web.ukonline.co.uk/george.gwilt/

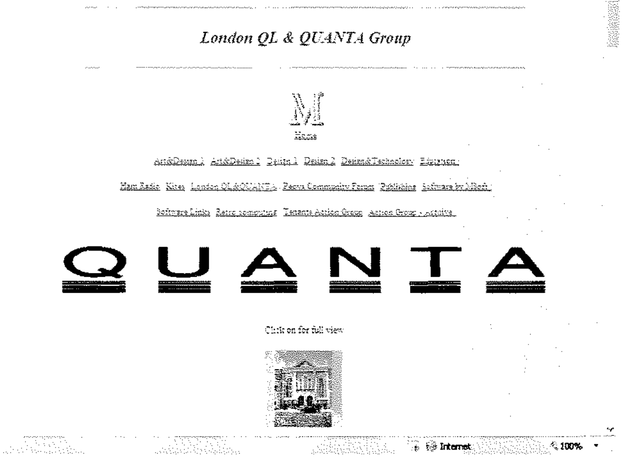
LONDON

Malcolm Cadman has announced an update of the London subgroup website:

"The London QL and Quanta Group web page has been updated..

A lot of new pictures of Quanta Workshops from recent years, have been added, as well as many minor changes.
http://www.mcad.demon.co.uk/lquan.htm

I have also begun a new web page for "Retro Computing", to list hardware, books, etc, that are available.
http://www.mcad.demon.co.uk/retro1.htm



London QL & QUANTA Group

Home

Art&Design 1 Art&Design 2 Design 1 Design 2 Design&Technology Education

Ham Radio Kites London QL&QUANTA Pecva Community Forum Publishing Software by MRsoft

Software Links Retro computing Tenants Action Group Action Group - Archive

QUANTA

Click on for full view

The website is a page dedicated to the QL on Malcolm's general website. Although fairly short it contains a wealth of information and links. These include a short history of the QL and a small amount of software to download. There are also links to Rick Dickinson's Sinclair designs which were featured in QL Today a year ago.

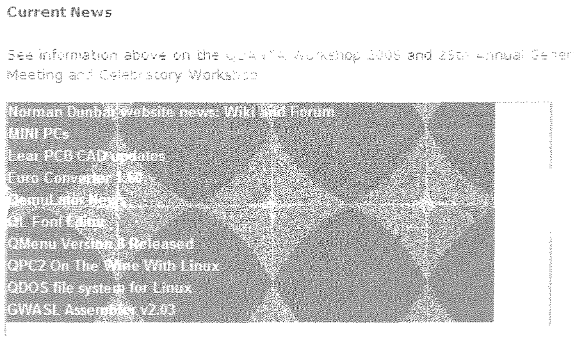
The retro page gives details of the QL books and hardware that the London subgroup has available for sale.

This is a good example of a simple site publicising all the facilities a subgroup can offer. Even a subgroup that did not wish to set up their own website could produce similar pages for the Quanta site. Will Manchester take up the challenge?

Malcolm tells us that the London subgroup may organise a QL show next year. No further shows are to be held in Byfleet or Portslade and should the show go ahead it will be the first Quanta show in the south for well over two years.

QUANTA

There have also been some changes on Quanta's main website. (www.quanta.org.uk). Readers of the Quanta magazine will know that Dilwyn Jones, Quanta's newly appointed news editor, has already made his mark. Now some of his news items are being posted on Quanta's website. This could become a valuable resource for the whole QL community with up to date news without QL-ers having to wait until the magazines



Current News

See information above on the QUANTA Workshop 2008 and 25th Annual General Meeting and Celebratory Workshop

Norman Dunbar website news: Wiki and Forum
MINI PCs
Lear PCB CAD updates
Euro Converter 1.60
Qemulator News
QL Font Editor
QMenu Version 8 Released
QPC2 On The Wine With Linux
QDOS file system for Linux
GWASL Assembler v2.03

are published. Dilwyn is also working hard to make Quanta's helpline more relevant for members, but it is not known if this will eventually have a place on the website.

Quanta has long had an aim to have public and members only areas of the website, but has yet to decide where the boundaries between the two should lie. This is a difficult decision for Quanta because there are several points of view with strong arguments for and against each.

Some argue that Quanta members have paid a subscription so that large areas of the website should remain closed to non-members. The Chairman took this attitude in a conversation with QL Today at the Manchester show.

A more intermediate view is that more content could be placed on the public area of the site, but only after it has first appeared in the Quanta Magazine. This would be fair to those members who do not have access to the internet. The acting magazine editor is believed to support this point of view.

The final view is that most information should be in the public area as a service to the whole QL community. In this way Quanta would demonstrate its relevance to the future survival of the QL and thus might attract new members.

In his Byts of Wood column in the last issue Roy Wood raised an interesting example of the problems that Quanta faces in this respect. Roger Godley has been updating the Psion suite to make use of GD2 colours and his work is probably the most important acquisition for the Quanta library in recent years. From the contact that QL Today had with Roger two years ago it was clear that he wanted Quanta to have the kudos for the project, but he would also like the software to be available to non members. It is strict Quanta policy that its library is only available to members, but, given the author's strong wishes, is this a case for an exception? Should his work have a place on the public area of the Quanta website?

In recent weeks there has been some effort to keep the Quanta website up to date although QL Today has had problems with out of date content on the website and in the Quanta Magazine. Quanta was slow in responding to our complaints in spite of the fact that similar problems occurred two years ago. QL Today believes that no advertising is better than out of date advertising and should these problems continue may be forced to review the current reciprocal agreement with Quanta.

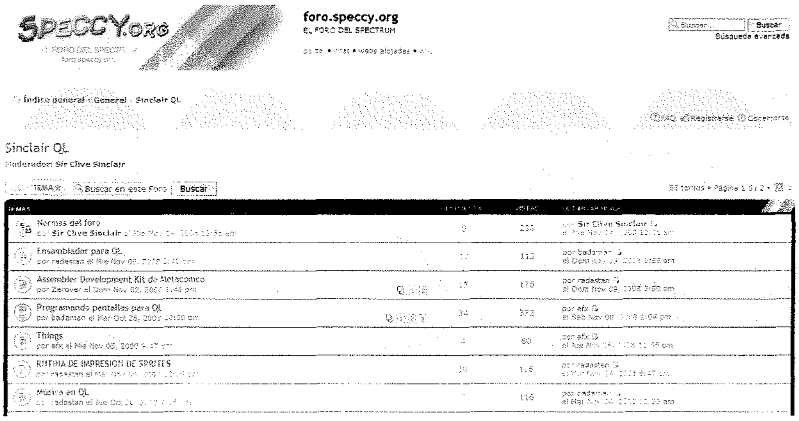
On the home page of its website, Quanta makes the bold statement "We have close links with international groups." Perhaps Quanta might like to name these groups together with the dates of the last contacts.

Spanish QL & Spectrum Forum

We have received the following news item via Quanta's news editor:

Javier Guerra writes:

"We have a new url for the Spanish QForum:
http://www.speccy.org/foro/viewforum.php?f=15
The old forum had a lot of spam."



foro.speccy.org
EL FORO DEL SPECTRUM

Buscar... Buscar
Búsqueda avanzada

Índice general ‹ General ‹ Sinclair QL

FAQ Registrarse Conectarse

Sinclair QL
Moderador: Sir Clive Sinclair

TEMA Buscar en este Foro Buscar

TEMAS	[illegible]	VISTAS	ÚLTIMO MENSAJE
Normas del foro por Sir Clive Sinclair [illegible]	0	235	por Sir Clive Sinclair [illegible]
Ensamblador para QL por radastan el Mie Nov 05, [illegible]	[illegible]	112	por badaman el Dom [illegible]
Assembler Development Kit de Metacomco por Zerover el Dom Nov 02, [illegible]	[illegible]	176	por radastan el Dom Nov 09, [illegible]
Programando pantallas para QL por badaman el Mar Oct 28, [illegible]	34	372	por afx el Sab Nov 08 [illegible]
Things por afx el Mie Nov 05, 2008 [illegible]	[illegible]	60	por afx el [illegible]
RUTINA DE IMPRESION DE SPRITES por radastan [illegible]	[illegible]	[illegible]	por radastan [illegible]
Música en QL [illegible] radastan [illegible]	[illegible]	116	por badaman el Mar [illegible]

QL2K

Jimmy Montesinos has made some changes to the help page of the QL2K website. He wrote that there had been some problems with the help page of QL2K. These were not problems with the QL2K itself, but with the website.

He has now made several changes to the website to solve these problems. This was not easy to do as there was a complicated situation involving the internet supplier, hosting services, firewall, network, software, routing, referencing and so on.

There were also severe demands on his time.

Jimmy expects that he will make further small changes in the near future but there will be no major malfunctioning of the site. He is also thinking of rebuilding the site to give it a more attractive design.

Jimmy has also given some advance information about the next release of the program. This will support 64 bits and Vista and will come in two editions, one for 32 bits and one for 64. There will also be some corrections to the sound support under Vista. If time permits he also intends to test a new version of QLAYT.

http://www.jadiam.org/QL

Failed Forum

In mid September Norman Dunbar set up an experimental forum on his website. In his own words:

"I'm experimenting a little with **http://qdosmsq.dunbar-it.co.uk** *and I have added a forum for discussion of all things QL, QDOS, SMSQ and so on. There's even an off topic area.*

At present, it is using over half of my allocated quota of disc, so it may not remain - it all depends. I can probably update my hosting account to get more space if I must."

In practice this forum proved not be a success with just a handful of people registering. Early in November the forum was taken down. One registered user commented that he found most of his needs catered for by the QL-users email group and this appears to have been the general opinion.

The QL-users group has mellowed in recent years, but in the past there have sometimes been virulent differences of opinion and heated discussion that put some QL-ers off. Prior to 2004 the then members of the Quanta committee expressed considerable distaste for the group seeing it as being immature and puerile. In November 2004 current chairman, John Mason, under the misapprehension that the Users group was a Quanta initiative, accused some of the users of abusing the "Quanta Committee's 'Freedom of Speech' tolerance policy" and threatened that Quanta could "invoke the constitution" to redress this.

Since that time Quanta's attitude to the group has softened. John Gilpin was the first Quanta officer to actively participate and John Mason now maintains a watching brief. Quanta's webmaster also monitors the group as part as his familiarisation with the QL community and this year an active participant in the group, Dilwyn Jones, has joined the Quanta committee.

The failure of Norman Dunbar's forum has wide reaching implications for Quanta as the organisation has long expressed a desire to have a forum for its own members. It is now debatable whether there would be sufficient interest in a Quanta forum.

Quanta has never made detailed plans for its forum or of how this would be moderated. The organisation has no tradition of encouraging vigorous debate and during the last year the present officers have shown a marked reluctance to allow disagreements and differences of opinion to be aired in the Quanta Magazine. When QL Today published an editorial that upset Quanta's chairman we offered Quanta a 2,000 word unedited right of reply. Sadly this offer was not taken up. Quanta has failed to give both Tony Firshman and QL Today's editor a right of reply in the Quanta Magazine to misleading remarks made about them by Quanta's officers. Quanta also failed to publish, give reasons for not publishing or even acknowledge a letter sent by email from the Birmingham subgroup giving their point of view about the misunderstanding that arose over last year's Midlands show (see QL Today V12 i2 p16).

Software Section

Once again Dilwyn Jones provides our software news:

QL Font Editor

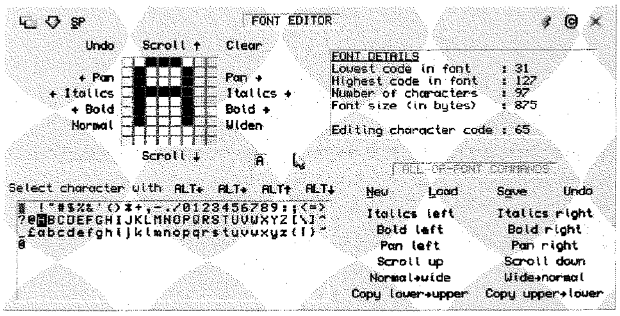
This is a new pointer driven font editor program from Dilwyn Jones. It lets you specify font character range as well as use the mouse (or cursor keys) to edit the characters. It has facilities to bolden or italicise individual characters, handle fat (8 pixel wide) fonts for CSIZE 1,0 use, pan/scroll characters around the editing grid and to copy upper/lower case ranges if you wish to do so.

The program was written in SBASIC using Easyptr 4 and compiled with QLiberator 3.36 to produce a program which runs on all WMAN2 systems. Expanded memory is needed, as well as Toolkit 2. The program will run on SMSQ/E systems, or QDOS with pointer environment version 2 or later. Several example fonts are supplied with the package.

QL Font Editor can be downloaded from **www.dilwyn.uk6.net/fonts/fonted2.zip**

It is a freeware program.

Elsewhere in this issue we start a new series of articles on fonts written by Dilwyn.



```
FONT DETAILS
Lowest code in font      : 31
Highest code in font     : 127
Number of characters     : 97
Font size (in bytes)     : 875

Editing character code : 65
```

```
Select character with ALT← ALT→ ALT↑ ALT↓
▒ !"#$%&'()*+,-./0123456789:;<=>
?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^
_£abcdefghijklmnopqrstuvwxyz{|}~
©
```

GWASL Assembler v2.03

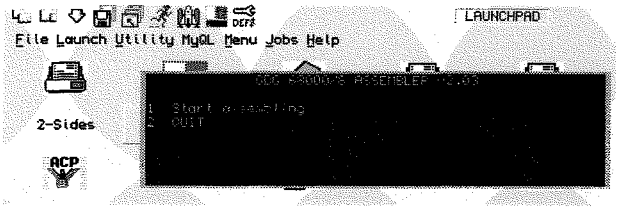
The latest version of Gwasl assembler by George Gwilt is now available from Dilwyn Jones' website. Gwasl is the Lite version of George's Gwass assembler, the main difference being that

Gwasl is suited for use on 68008 processor systems (i.e. a standard QL and some emulators) whereas the full Gwass assembler is designed to use the additional instructions present on later 680x0 systems. Gwasl is especially valuable for those following Norman Dunbar's Programming In Assembler series in QL Today magazine.

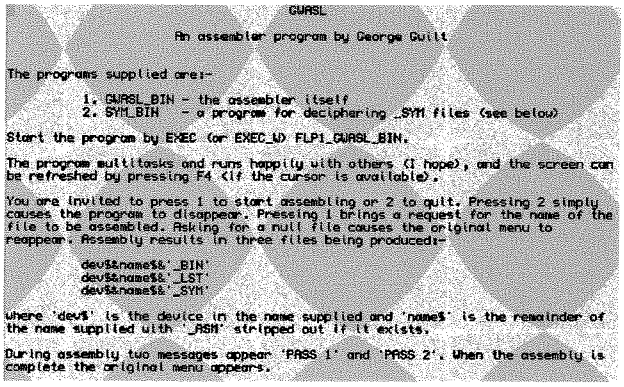
Changes listed for recent versions include TABs can be used as well as SPACEs and the maximum number of lines and labels is now configurable.

The program and its source code files can be downloaded from:

www.dilwyn.uk6.net/asm/index.html



```
1  Start assembling
2  QUIT
```



```
                              GWASL

              An assembler program by George Gwilt

The programs supplied are:-

        1. GWASL_BIN - the assembler itself
        2. SYM_BIN   - a program for deciphering _SYM files (see below)

Start the program by EXEC (or EXEC_W) FLP1_GWASL_BIN.

The program multitasks and runs happily with others (I hope), and the screen can
be refreshed by pressing F4 (if the cursor is available).

You are invited to press 1 to start assembling or 2 to quit. Pressing 2 simply
causes the program to disappear. Pressing 1 brings a request for the name of the
file to be assembled. Asking for a null file causes the original menu to
reappear. Assembly results in three files being produced:-

        dev$&name$&'_BIN'
        dev$&name$&'_LST'
        dev$&name$&'_SYM'

where 'dev$' is the device in the name supplied and 'name$' is the remainder of
the name supplied with '_ASM' stripped out if it exists.

During assembly two messages appear 'PASS 1' and 'PASS 2'. When the assembly is
complete the original menu appears.
```

LEAR PCBCAD

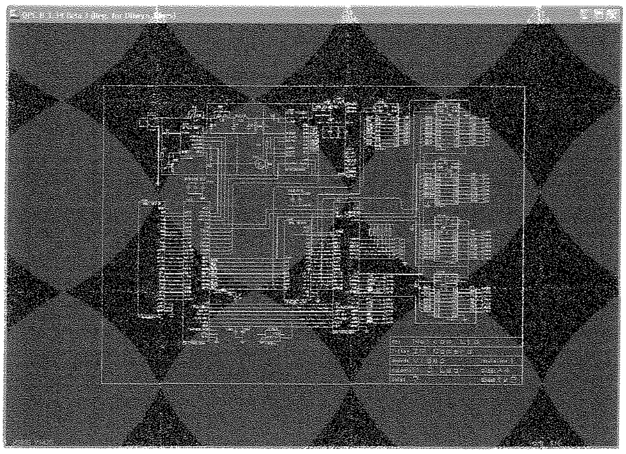
Malcolm Lear has excelled himself as this time we have two upgrades to his program. QL Today is still looking for a reader willing to review the software.

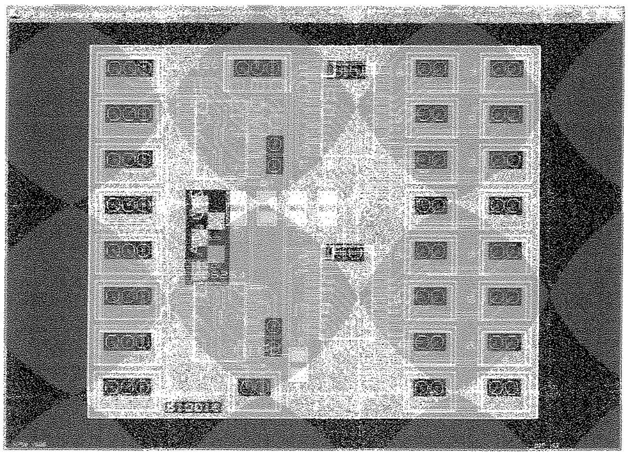
VERSION 6.53

Malcolm Lear has released v6.53 of his PCB Cad program, to include some important changes to bring back old QL compatibility. He's not had a QL for more than 10 years, so he is now started using the QL2K emulator to ensure correct QL operation. It turns out that the GWASS assembler has been using 68040 instructions, so he has gone back to using the GST program.

Here is a list of changes in version 6.53, taken from the REVISIONS.TXT file in the package:

NC text drill file shows drill sizes in both metric and imperial. SMD device notification in placement file. Using QL2K to check QL compatibility. 68040 op codes removed. QL users may have had problems since 6.22. Moved back to GST assembler. Memory allocation routines rewritten to be more QL friendly. QL's must have full memory 768K expansion to run program. QL will not run any hi-colour commands. File loading problems with QL2K which may well be emulator issues.





VERSION 6.59

Malcolm Lear released v6.59 of his PCB Cad program on 4th November 2008. This version has now added the facility to draw circles and ellipses. The program has been updated several times recently, so here's a list of major changes in recent versions of the program, taken from the REVISIONS_TXT file supplied with the program:

6.56 02-10-08 Logical layer mode bug in dot matrix layer selection fixed. Cadtk.bin and Printer.bin merged to save memory. All automated file format updating now done in assembler. File loader now removes duplicate information blocks.

6.57 Mode 8 graphics aspect ratio back to pre 6.22 while retaining changes needed for future pointer environment version. This produces much cleaner diagonals and gives the impression of higher resolution. Screen mode toggle now disabled when hi-colour drivers detected. More dot matrix issues resolved. Invisible pads and blocks layers in logical layer mode are now invisible.

6.58 Element stats at bottom left of screen now include layer information. Grid step and scale updated when component picked up or dropped.

6.59 04-11-08 Circle and ellipses can now be drawn using tracks. Extension names changed to match style use in the SBasic program.

TURBOPTR

George Gwilt writes:

Remarks by Dilwyn Jones at the last QUANTA AGM (thanks Dilwyn) caused me to make TurboPTR more user-friendly. The result is now available at

http://web.ukonline.co.uk/george.gwilt/

Goodbye Eindhoven?

A long-standing QL tradition may have come to an end when the last show of 2008 was held at the St. Joris College in Eindhoven in October. No other show location has been used for QL events so long and so frequently.

A full report appears elsewhere in this issue.

Second Thoughts

Some users are having doubts about the suitability of the Asus eeePC for QL use. Per Witte reported on the QL Users email group:

"After using the eeePC 701 for a while, I gave it up as too limiting for my purposes. The battery life is crap, the fan whirrs incessantly because the thing produces far too much heat. It also seems stupid to put up with a 7" screen when the box is more like 10."

Richard Kilpatrick responded:

"I agree that the 701 is too limiting; this is why I favour the Acer Aspire with 120GB HD. It's £229, so still well within the cheap end of the spectrum (compared to £300+ for the 901/MSI Wind) and is very small and efficient.

Do try one. The 1024 x 600 resolution over the Eee 701's 800 x 480 is a massive advantage."

In the last issue Jochen Merz confirmed that QPC runs well on the Asus Wind.

Rewriting History

It is a myth that the QL is a Sinclair product. In fact the first QL was produced in 1909 long before Sir Clive was born. We are not sure what was used for a display because televisions and monitors had not been invented, but you know the QL is almost 100 years old because you read it in QL Today.

Red faces all around the QL Today editorial office as two eagle eyed proof readers failed to notice a missing word in an editorial that made the QL four times older than it is. We said that 2009 was the QL's centenary when we meant its quarter centenary.

Quanta has announced some details of its quarter centenary celebration to be held next year. The venue is the Allesley Hotel, which is about 3 miles north of Coventry city centre.

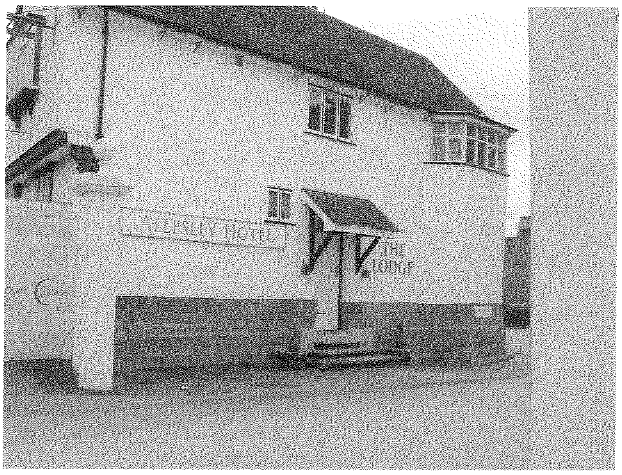
The event will be held on Saturday 18th and Sunday 19th April. Quanta is promising a program of talks, an interactive problem solving workshop, a repair desk and traders. There will be a celebratory dinner on the Saturday evening and the Annual General Meeting on the Sunday.

One person who will not be attending is Tony Firshman who has a prior singing engagement in Venice (moonlighting as a gondolier?). Geoff Wicks has also announced that he will not be playing an active part in the show, nor will he set up a Just Words! table:

"Quanta cannot have it both ways. They cannot criticise me in the Quanta Magazine, fail to offer a right of reply and then expect me to work for them. Also Quanta showed a lack of gratitude for the work I did to ensure the success of the Solihull show last year. The chairman pointedly omitted me from the list of people he publicly thanked, which I find to be unbelievably mean spirited."

An activity over which there is a question mark is some form of internet connection. A Skype connection was used at the Italian show for a presentation on QemuLator, but nothing of this type has ever been attempted at a UK show. Dilwyn Jones is known to be actively investigating how Skype could be used by Quanta in both shows and elsewhere. However the hotel has only limited internet facilities.

QL Today has some bad news for drivers who needed a subsidy of about £25 per car for AA sign posting to find their way to QL is 21 in Portsmouth. The Allesley is even more difficult to find and there is a nasty roundabout to negotiate just before the hotel. As far as QL Today could see there are no signposts for the hotel itself. However once you are in Birmingham Road you cannot miss the Allesley as our photo of the entrance shows. Motorists are advised to avoid the centre of Coventry if possible as the inner ring road is small but can be confusing.



Neither Quanta nor the hotel can provide advice for public transport users, but QL Today can. A bus stop to Coventry is directly opposite the hotel entrance and the bus stop from Coventry about 20 yards up the road. The service is fairly regular and the stops are serviced by buses 7 (not Sunday), 75 and 900. However in Coventry city centre you are on your own. Even QL Today could not trace the nearest bus stops to the railway station for these services.

We advise you to try the website:

www.travelcoventry.co.uk/bus/index.asp

This gives details of bus timetables and they are promising maps of the bus stops before long.

QUANTA Crisis

It is not only QL Today that has had problems getting its magazine to the readers on time. The June/July Quanta Magazine was published at the end of August and since then there has been no further issue. A formal announcement from Quanta on the QL users email group explains the situation:

"Due to not receiving any response to communications sent to our new Magazine Editor - Tony Hill - over a now considerable period, this committee have asked John Gilpin - QUANTA Treasurer and Membership Secretary - to stand in again as Joint Magazine editor until (at least) the AGM in April 2009. The other Joint Editor's position will remain with John Mason.

In order to notify you all of details for the forthcoming Celebration Workshop and AGM in April 2009, John G has agreed to publish Quanta Magazines for August/September/October 2008, (a three month's issue), and November/December 2008/January 2009 (another three month's issue). This will bring us up to the end of Volume 25. He has also agreed to publish a Magazine for February/March 2009 which will be our "Celebration Issue" - QUANTA will be 25 years old in February 2009."

Behind this statement is a human story that QL Today discovered when some of our emails to John Gilpin bounced back. In addition to Quanta problems John has had similar problems with the scout movement when he had to suddenly take back duties that he had passed on to someone else. To add to his problems his internet server made a mess of upgrading his broadband with among other things emails going astray.

To use a Dutch phrase John is "Having to mop up with the tap still running".

John Gilpin is the linchpin of Quanta and without him Quanta would probably cease to function. His duties as treasurer require some knowledge of accountancy and as acting Newsletter Editor a complete familiarity with Page Plus. In a sense he is irreplaceable unless someone else has or is prepared to learn these skills. Through his active presence on the QL Users email group John Gilpin has become the human face of Quanta to whom many first turn to for help in Quanta matters.

In John's own words:

"The situation seems to be ongoing and if QUANTA Members are to receive their Magazines on time and of a reasonable quality, a new Magazine Editor MUST BE FOUND by April 2009 as I cannot guarantee my time to continue to do this job after that date."

Special Characters using QPCPrint

by George Gwilt

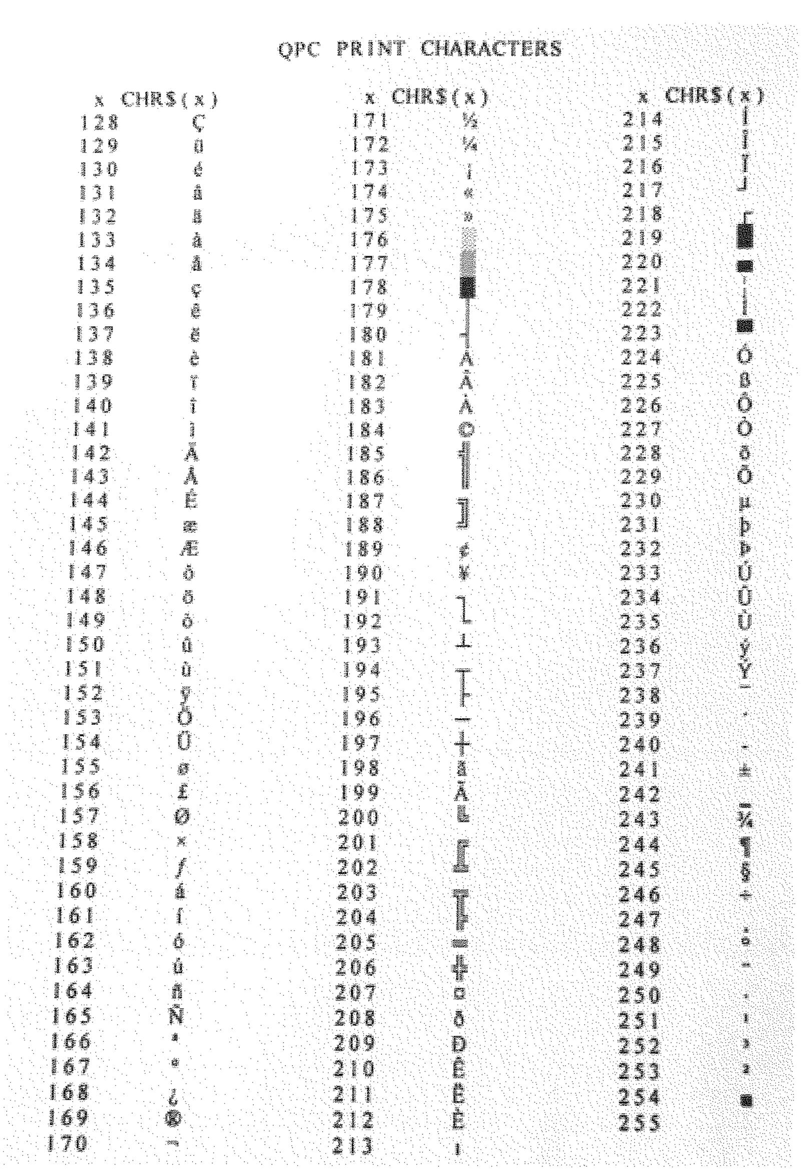
One of the most liberating things I have come across is QPC Print. The other is of course QPC2. All the old dot matrix printers I used in the past with various QLs ranging from Super Gold Card to a Q60 have given up the ghost. Since I use Archive and Perfection for printing, the demise of the old printers was a severe blow ... until QPC Print came along.

Pound Sign and Hash

Perhaps you remember the days when articles would appear with ` being printed instead of ♯. Eventually most people found out that it was a good idea to print using some sort of editor which had a printer driver. In this the instruction was set to replace `, (CHR$(96), by the instruction to the printer to use the UK character set followed by the code for ♯, CHR$(35), followed by a return to the US code. A hash was simply sent as itself.

In the days when I used to send output directly to the printer I used the QL's translation facility to alter ` to the three characters L, BS (Back Space), =. This gives a serviceable, though not very elegant, version of the pound sign.

How nice it would have been to send just one character value to the printer for `. QPC Print makes this possible.



QPC PRINT CHARACTERS

x	CHR$(x)	x	CHR$(x)	x	CHR$(x)
128	Ç	171	½	214	Í
129	ü	172	¼	215	Î
130	é	173	¡	216	Ï
131	â	174	«	217	┘
132	ä	175	»	218	┌
133	à	176	░	219	█
134	å	177	▒	220	▄
135	ç	178	▓	221	¦
136	ê	179	│	222	Ì
137	ë	180	┤	223	▀
138	è	181	Á	224	Ó
139	ï	182	Â	225	ß
140	î	183	À	226	Ô
141	ì	184	©	227	Ò
142	Ä	185	╣	228	õ
143	Å	186	║	229	Õ
144	É	187	╗	230	µ
145	æ	188	╝	231	þ
146	Æ	189	¢	232	Þ
147	ô	190	¥	233	Ú
148	ö	191	┐	234	Û
149	ò	192	└	235	Ù
150	û	193	┴	236	ý
151	ù	194	┬	237	Ý
152	ÿ	195	├	238	¯
153	Ö	196	─	239	´
154	Ü	197	┼	240	-
155	ø	198	ã	241	±
156	£	199	Ã	242	‗
157	Ø	200	╚	243	¾
158	×	201	╔	244	¶
159	ƒ	202	╩	245	§
160	á	203	╦	246	÷
161	í	204	╠	247	¸
162	ó	205	═	248	°
163	ú	206	╬	249	¨
164	ñ	207	¤	250	·
165	Ñ	208	ð	251	¹
166	ª	209	Ð	252	³
167	º	210	Ê	253	²
168	¿	211	Ë	254	■
169	®	212	È	255	
170	¬	213	ı		

Boxes

Some time ago I wrote a program for Archive to keep track of the contents of wine racks. As a check on the accuracy of the record I used a second program which printed a representation of each rack. Each of the spaces contained either a blank or a character depending on whether the record showed that space containing a bottle or not. By comparing the representation with reality I was able to detect, and so correct, anomalies.

The grid representing a wine rack was made up of eleven characters available as special graphics characters on my Epson printers. The characters were:

A horizontal line bisecting the character rectangle. A vertical line bisecting the character rectangle. Four corners each with the point in the centre of the character rectangle and isolating a quarter of it. The five distinct figures produced from the corners taken in pairs.

As an example three lines each of three characters suffices to produce a boxed character. The first line is a top left corner, then a horizontal line followed by a top right corner. The second line is two vertical lines with the character in between. The third line is a bottom left corner, a horizontal line and then a bottom right corner.

When my printers gave out I decided, of course, to use QPC Print. I was delighted to discover that printer output from Perfection and Archive was accepted easily by

QPC Print. All, that is, except certain special characters, including the box elements. I had laboriously re-written my Archive box program to produce boxes made up of (I think) plus signs. The result was usable but nasty. It then suddenly struck me (meaning that my subconscious, which had been mulling over the problem at leisure, eventually decided to alert my consciousness and tell it...) that you could send character values 128 to 255 to QPC Print and have real characters printing. Investigation showed that the eleven box characters are included. They have values 179, 180, 191 to 197 inclusive, 217 and 218.

This means that I can produce my boxes via Archive using QPC Print. I may say that the alteration to my original program to achieve this was pretty easy.

I also altered Perfection's printer driver so that it would be possible to put into a document many of the special characters in the QL fonts and also have them printed via QPC Print. One example of this is the greek lower case beta. In particular the pound sign has value 156 in QPC Print.

I could describe the remaining 117 characters in the range 128 to 255 which QPC Print produces, but it would be a lenghty task and not very useful. Instead I show the characters in a table on the previous page.

Resizing Windows

by Dilwyn Jones

In response to George Gwilt and Jochen Merz's items about resizing windows in the last QL Today, here are some hints and a listing to show a simple way to make window resizing behave a bit more predictably in Easyptr 4.

It might work in earlier versions of Easyptr, using Per Witte's resizing routines from his Knoware website (not tried).

Many of my early programs written using Easyptr 4 managed to resize windows, but the new origin position was very hard to predict because, as was said, PE routines do not guarantee menus will reappear at the required point. Resizing was a bit random - you never really knew where the new size menu would appear when drawn.

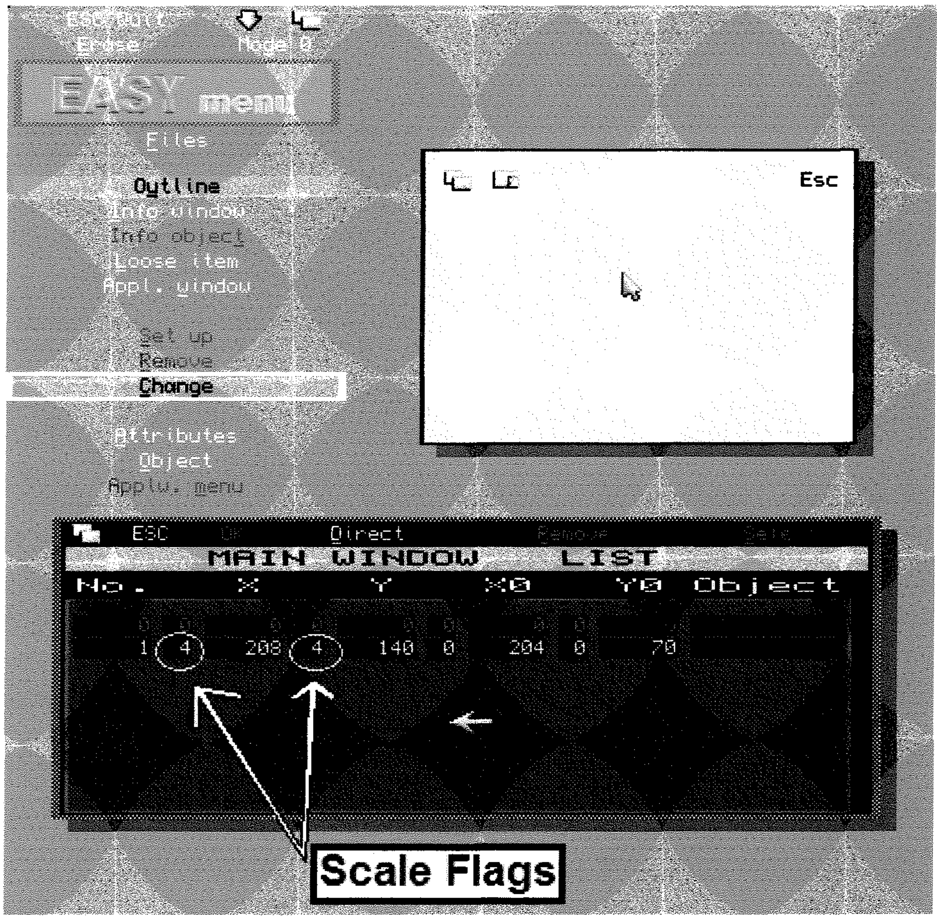
This demonstration routine written in SBASIC seeks to provide a little routine that behaves rather better than a simple resize might achieve, by reading window sizes and origins before the resize, so we can cheat a little and adjust the results to compensate.

First, we need to design a small menu window in Easymenu 4. This has just three loose items, numbered -1, -2 and -3:

-1 is the top left corner MOVE icon
-2 is the RESIZE icon next to it
-3 is a simple ESC item to quit from the program.

Make sure that once you have designed the window, that you set the scale flag (see figure 1) values to 4 each for width and height, and 0 each for x and y origin.

Figure 1 - Setting scale flag values to 4



```
MAIN WINDOW    LIST
No.     X         Y        X0        Y0   Object
  1  4   208  4   140  0   204  0    70
```

Next, add the three loose items as shown in figure 1. I've used the system icons for MOVE and RESIZE, the double box symbols, although you can design your own if you wish.
It is important that the top left corner icon is loose item -1, that the next is loose item -2, and the Esc icon at the top right is loose item -3 to match what the example listing expects.
When it comes to assigning key presses to the loose item, set ASCII code 5 or 245 (CTRL F4) for the MOVE item, 6 or 241 (CTRL F3) for RESIZE and 27 for the Esc item. 5 and 6 are "internal" system codes, while 241 and 245 are standard keypress ASCII codes.

Then, once you have saved the menu with the name 'resizer', you can type in and run the following program:

Listing 1 - Resizer_bas

```
100 MDRAW #0,"flp1_resizer_men"
110 REPeat program
120   num = MCALL(#0)
130   SELect ON num
140     =-1 : WMOV #0,-1 : REMark Move
150     =-2 : REMark Resize
160       DIM pr%(16) : PVAL #0,pr% : REMark get pointer request details before
          resize in array pr%()
170       WSIZE #0,xv%,yv% : REMark call resizing routine, get x and y variation
180       MCLEAR #0 : REMark clear old menu
190       REMark new size
200       xsize% = pr%(8) + xv% : REMark vary horizontal size
210       ysize% = pr%(9) + yv% : REMark vary vertical size
220       MDRAW #0,"flp1_resizer_men",pr%(10),pr%(11),xsize%,ysize%
230     =-3 : EXIT program : REMark Esc
240   END SELect
250   stat% = MSTAT%(#0,num TO 0) : REMark reset loose item status
260 END REPeat program
270 MCLEAR #0 : INK #0,7 : REMark reset after use
```

You should find that as you click on the second loose item (RESIZE) the code in lines 150 to 220 take care of resizing and redrawing the menu, with the origin locked at the same position as before the resize where possible.

Line 100 draws the menu initially.

Line 120 calls the menu reading routine MCALL.

Line 140 handles the MOVE routine. Note that if you set a key code of 5 for MOVE, this is handled internally and this command is never called. WMOV #0,-1 is a special version of WMOV that handles MOVE calls in Window Manager 2.

Line 160 is the "brains" behind the resizing. the PVAL command does a read of the window details before the call to resize. In other words, it stores the origin and dimensions of the window before the resizing, so that we can use this to fix the origin after setting the resize.

PVAL returns a 16 element array, as described under the PVAL entry in the Easyptr manual. The elements of pr%() we are interested in are:

pr%(8)	window width
pr%(9)	window height
pr%(10)	window x-origin
pr%(11)	window y-origin

Line 170 calls the WSIZE command, a new command in version 4 of Easyptr. It takes 3 parameters, a channel number, and two variables which return the variation or change in size of the menu after resizing. In other words, if it was 200 pixels wide beforehand, 300 pixels after resizing, xv% would contain 100 after the call.

After resizing, the MCLEAR statement in line 180 clears the old menu size ready for it to be redrawn in the new size.

Lines 200 and 210 work out the new size of the menu by adding the change in size to the original size.

Line 220 draws the menu in the new size. Note that this time we use the extended version of the MDRAW command which allows size and location details to be specified:

```
MDRAW #channel,menu_name,x_position,
       y_position,xsize,ysize
```

We use pr%(10) for x_position and pr%(11) for y_position so that the menu's origin stays in the same position where possible (unless the new size means the menu falls outside the visible screen, in which case details may be automatically adjusted by the system).

xsize% and ysize% have been adjusted by now to specify the new menu size.

Line 230 detects the ESC keypress and jumps out of the loop called "program", and the MCLEAR statement in line 270 removes the menu as the program stops.

A Continental Puzzle

by Geoff Wicks

Two difficult questions to test your QL knowledge:

1) Quanta has always been the largest QL user group, but which land hosted the second largest group, being at its peek 1,200 strong?
2) How come that user group did not have a single member?

I doubt if many people would get the first question right. The correct answer is the Netherlands and Stichting Sin_QL_Air, although other lands might like to challenge this. You may have guessed the second question is a trick one, because to answer it you would need a knowledge of Dutch law.

The Dutch have two forms of organisation a "stichting" and a "vereniging". The latter has a tighter legal structure and members. The former has a looser legal structure and no members. Its supporters are known as donors.

I was a Sin_QL_Air donor from 1989 to 1996, but even I was surprised to learn from Sjef van de Molengraaf that at one time there were 1,200 donors. Given that Quanta had about 2,000 members at its peak and that the Netherlands has a population a quarter that of the UK, the Netherlands can probably claim, per capita, to have been the most QL friendly land.

Sin_QL_Air and Quanta are the only active user groups still in existence. In recent years Sin_QL_Air has hosted three shows a year in Eindhoven, but this could shortly change.

The Dutch have always played an active role in the QL community. Formerly they held 6 shows a year, one of which was an international event that was a "must" for anyone who was anyone in the QL community. Shows have become less frequent and international shows smaller over the years, but in 2004 Sin_QL_Air hosted the most significant international QL event in the last 10 years. This brought most of the QL developers together and gave a boost to what was then a flagging QL community. To remind you of that event we print a picture of the after-show dinner taken by Tony Firshman on the cover.

One Sin_QL_Air donor, Wolfgang Uhlig, wrote a great deal of software for the QL. His first program was a commercial database, SuQcess and he then wrote a number of freeware tools to assist QL programmers in using GD2 colours. When Wolfgang lost interest in the QL, Bob Spelten took over his work and his updates of the programs regularly feature in QL Today's news pages.

Bob works for an international computer company and by coincidence when I first lived in the Netherlands and was still learning Dutch, I worked for two and a half years in the Transport Department of the same company. I gained two things from this work. The first was learning to drive a fork lift truck but the second is more dubious. Like most native UK English speakers I speak Dutch with a heavy accent, but the experts tell me I speak some words with a pronounced coarse Amsterdam working class accent.

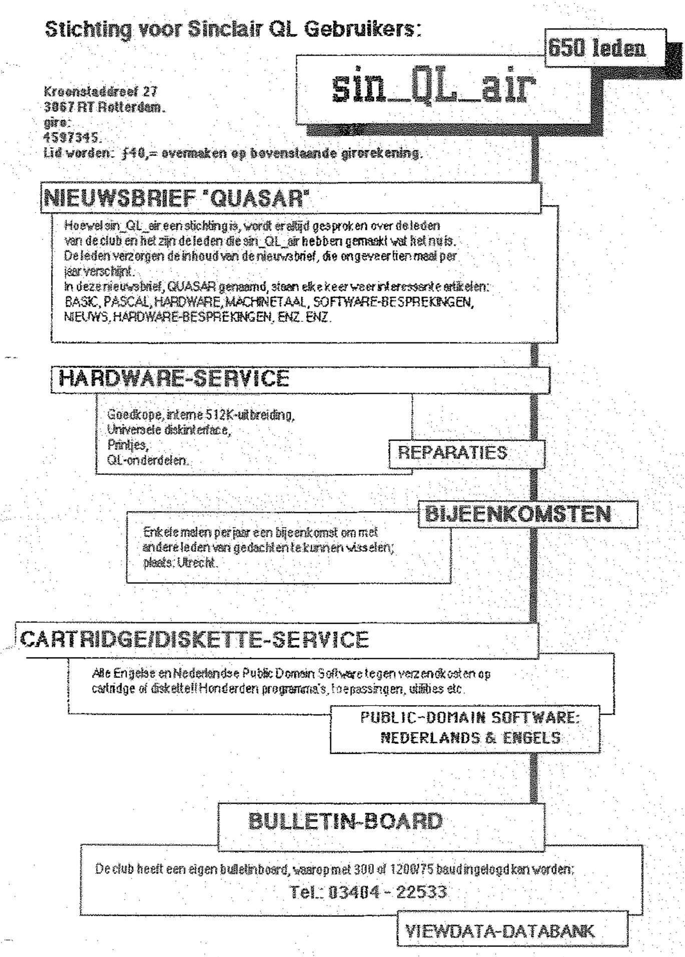
It was while I was at the company that a revolutionary new product was launched called a "wordprocessor". We even set up a special room to demonstrate it to potential customers. Strangely it had a screen in A4 portrait format and in place of the then usual green text on a black background had black text on a white background. No doubt this was done to help a computer phobic secretary think it was still a sheet of paper.

I cast jealous eyes on the product thinking "I wouldn't mind one of those". The Head of Supplies noted my keen interest and strictly forbade me to touch it. Little did he or I know that within four years I would own a wordprocessor more in my price bracket. It was called Tasword Two and ran on a Spectrum.

Just a few years further on and Sin_QL_Air would come into being with regular shows held at the St. Joris College in Eindhoven.

The first QL meeting in the Netherlands was held in the Summer of 1984, and it is a little known fact that there have been two Sin_QL_Airs. The first formally came into being in March 1986 and was closed on 1st April 1992. The second was formed just two days later.

Sin_QL_Air was an active user group and took part in what was then the Netherlands most important annual computer show held in one of Amsterdam's main exhibition halls. It was at such a show that I first heard of the stichting. Our illustration shows a recruiting poster used at the 1987 fair. The organisation also had its own magazine, Quasar, although this ceased publication in 1997 because of a lack of copy. In recent years no donations from supporters have been requested as the cost of running the shows was low.



Stichting voor Sinclair QL Gebruikers:

650 leden

sin_QL_air

Kroonstaddreef 27
3067 RT Rotterdam.
giro:
4597345.
Lid worden: ƒ40,= overmaken op bovenstaande girorekening.

NIEUWSBRIEF "QUASAR"

Hoewel sin_QL_air een stichting is, wordt er altijd gesproken over de leden van de club en het zijn de leden die sin_QL_air hebben gemaakt wat het nu is. De leden verzorgen de inhoud van de nieuwsbrief, die ongeveer tien maal per jaar verschijnt.
In deze nieuwsbrief, QUASAR genaamd, staan elke keer weer interessante artikelen: BASIC, PASCAL, HARDWARE, MACHINETAAL, SOFTWARE-BESPREKINGEN, NIEUWS, HARDWARE-BESPREKINGEN, ENZ. ENZ.

HARDWARE-SERVICE

Goedkope, interne 512K-uitbreiding,
Universele diskinterface,
Printjes,
QL-onderdelen.

REPARATIES

BIJEENKOMSTEN

Enkele malen per jaar een bijeenkomst om met andere leden van gedachten te kunnen wisselen; plaats: Utrecht.

CARTRIDGE/DISKETTE-SERVICE

Alle Engelse en Nederlandse Public Domain Software tegen verzendkosten op cartridge of diskette!! Honderden programma's, toepassingen, utilities etc.

PUBLIC-DOMAIN SOFTWARE:
NEDERLANDS & ENGELS

BULLETIN-BOARD

De club heeft een eigen bulletinboard, waarop met 300 of 1200/75 baud ingelogd kan worden:
Tel.: 03404 - 22533

VIEWDATA-DATABANK

No location has been used by QL-ers for so long or so frequently as the St. Joris College, but all good things must come to an end.

The school was used because two QL-ers, Sjef van de Molengraaf and Karl Boekema worked there. They were allowed to use the main hall free of charge together with many of the school's facilities. Sjef retired 6 years ago and slowly he has become a lesser known figure as new staff have been appointed. There was a warning on the wall at the time of QL2004 when a new school head allowed us to use the school, but not its computer projectors or internet facilities. Since then a second new head has come to the school and he is now reluctant to allow Sin_QL_Air to meet on the premises.

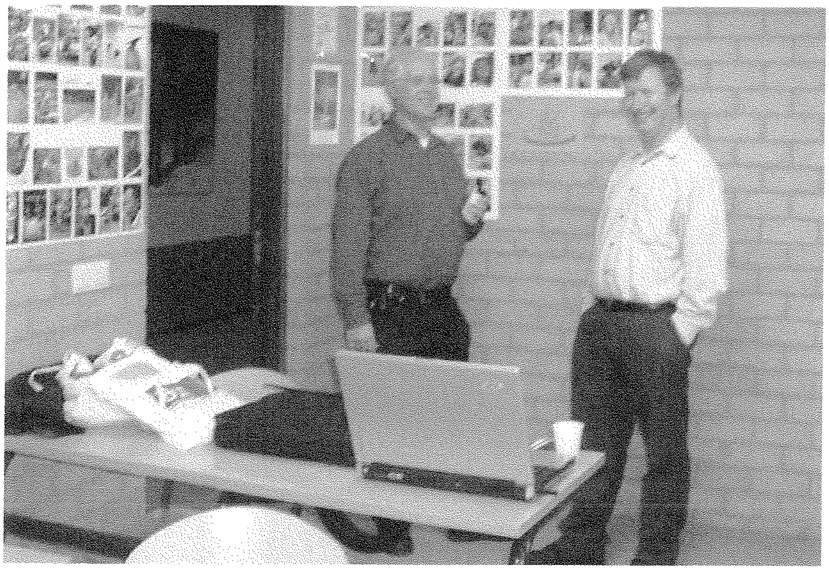
Sin_QL_Air now has a difficult choice to make. One possibility is to meet at a more central location such as Utrecht, but this will mean hiring a hall at commercial rates. Would the numbers attending justify this? One advantage of the Netherlands is that it is within easy reach of several European countries and ideal for international shows, but is there interest in this? QL2004 was a great success, but there was little interest for an international show in 2005. There has also been no response to QL Today's query about a possible continental celebration of the QL's quarter centenary next year.

The other possibility is for the organisation to be wound up, but here there is also a complication. Like Quanta, Sin_QL_Air has built up a large capital from the days when it was a much larger organisation. If Sin_QL_Air is wound up its constitution states that the money has to be used for the development of the QL. This is more difficult than you might think.

When Sin_QL_Air met at Eindhoven for the last time this year we were banished from the main hall because a group was meeting there who had paid for it. It was a small group that gathered there and I was only able to make it for the last hour. It was, however, an international group with representatives from the Netherlands, Belgium, Germany and the UK.

Eindhoven may soon be no more, but fond memories of it will remain not only in the Netherlands but in many other countries.

A few impressions from the last meeting at Eindhoven (not in the main hall), October 2008



Postscript for Non-UK Readers:
We would welcome more contributions from our non-UK readers. Remember that the editor knows first hand how difficult it can be to write something in a language that is not your native tongue. I shall be more than happy to improve and rewrite a text and then send you a copy of the revised version for your approval.
One of my greatest achievements in the Netherlands was to have an article published in a national newspaper. Late in the afternoon the day before I was telephoned to give permission for small changes to be made to improve the Dutch. Much to my surprise the next day I discovered my article was being used as background to their lead news story and was advertised on the front page.
I was glad I dared to write it and thankful for the changes that were made.

Configuration Block Utilities (Part 1)

by David Denham

Some time ago, I wrote about the Config programs and the use of Configuration blocks. In this article I intend to look at several free utility programs you can get for creating configuration blocks and compare them to see which might be best for your needs.

The programs I'll be looking at are:

Basconfig – the original program from Oliver Fink, later updated by Norman Dunbar, Ewald Ikeman, Dilwyn Jones and Rich Mellor.

Q-Config – a more recent and significantly enhanced version of Basconfig from Rich Mellor.

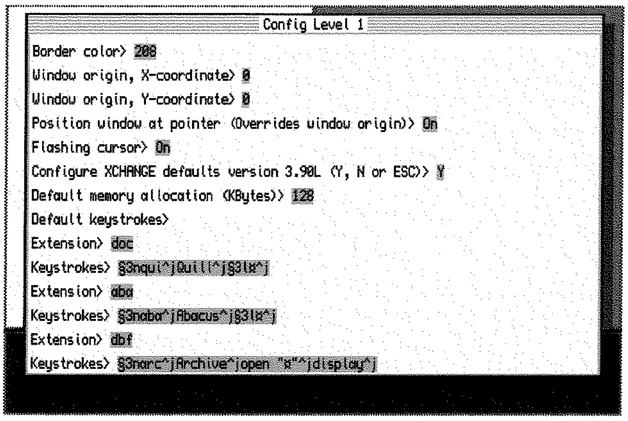
U-Config – George Gwilt program, a successor to the earlier T–Config.

FileCfg – Timothy Swenson's add–on for Basconfig

What Is A Config Block?

It's a system whereby you can include data telling a program how to start up - the data could be a default drive, for example, or the name of the directory where the program lives and finds its files, the colour of ink or paper the program uses to display itself on the screen, whether the program turns sounds such as error message tones off or on and so on.

This data is built into the code of the program and includes a list of options which are available to you. It is structured in a standard format, which allows a program such as Config or Jochen Merz's Menuconfig program to ask you what to change and to alter this data in the program, so that the next time you start the program, it will use the startup default values you have set. When a program like Config asks you questions to change the values of these startup defaults, it actually patches the executable program itself to make these changes "permanent".



```
Config Level 1
Border color> 208
Window origin, X-coordinate> 0
Window origin, Y-coordinate> 0
Position window at pointer (Overrides window origin)> On
Flashing cursor> On
Configure XCHANGE defaults version 3.90L (Y, N or ESC)> Y
Default memory allocation (KBytes)> 128
Default keystrokes>
Extension> doc
Keystrokes> §3nqui^jQuill^j§3l*^j
Extension> aba
Keystrokes> §3naba^jAbacus^j§3l*^j
Extension> dbf
Keystrokes> §3narc^jArchive^jopen "*"^jdisplay^j
```

Figure 1: The Config program, configuring Xchange

It is one thing to insert these bits of data into a program and to change them with another program. Your program must itself be able to read this information. Most utilities to create configuration blocks provide some mechanism such as basic extensions for the program to be able to read these bits of data. Bear in mind that there are several types of data which can be included in Config blocks, such as strings, 3 lengths of integer (byte, word, long word), character, code and select. There are also two levels of Config data, known as Level 1 (the older type, which is simpler to handle) and Level 2 (a more recent development which although harder to program, provides additional facilities).

Where To Get These Programs

I got copies of all of mine from Dilwyn Jones's website, though they are all freeware or PD, so you should be able to get copies from PD libraries and CD compilations too as well as most websites offering free QL software, such as Thierry Godefroy's website

Thierry Godefroy: **http://qdos.dyns.net/**
Dilwyn Jones: **http://www.dilwyn.uk.net/index.html**

Basconfig

This was the original program by Oliver Fink of QLaboratories in Germany many years ago. Although Oliver no longer updates this particular original version, it has been updated by several people since, with some bug fixes and additional facilities included along the way. It asks you a succession of questions such as how many items to include, what data type you require, ranges for the data (e.g. string lengths or limits on number values), what prompts to use and so on. This is then bolted onto the end of a set of basic extensions to return the data to the program as strings, integers, code values etc. This data block can then be LRESPR'ed for use while developing an interpreted basic program, and included in a compiled program with a $$asmb directive for QLiberator compiled programs, or a %% directive for Turbo compiled program. Basconfig can only create config blocks for interpreted or compiled SuperBasic or SBasic programs, it can't create blocks for assembler or C programs.

While this is pretty easy to do, it can be time consuming and you have to make sure you get it right first time, because Basconfig cannot edit its own config blocks once saved. Not a problem for simple one or two item blocks, which are easy enough to redo from scratch, but a bit of a chore for a larger config block with many items.

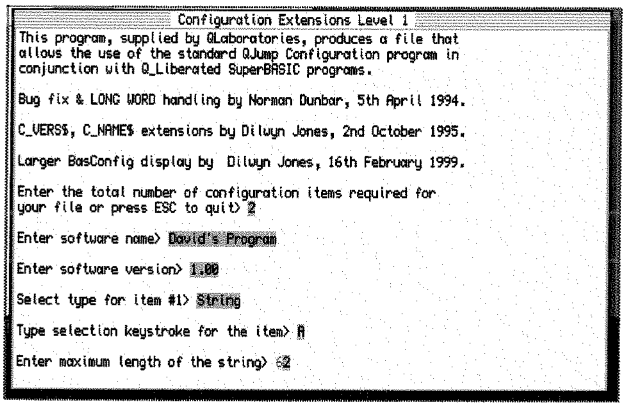
When you get a copy of Basconfig, make sure you get a fairly recent version and also note that somewhere among all the updates to the program, the version numbers got a bit out of sequence between authors. Versions 1.12, 1.12a and 1.13 all seem OK and have their own particular features and issues. Most of these versions have extensions to return the program name and version number from the config block, as well as long word type handling which Oliver's original version did not. You can see what changes were made in which version by reading the Quill document called bsconfig_doc which is part of the package. Most versions of Basconfig seem to include the source code, so you could if you needed to modify it to fix bugs or add more facilities.

The main parts of Basconfig are:

- Basconfig_obj or _rtm – the main part of the program, which guides you through the process and generates the finished config block. In most versions, it looks like the Config program itself.
- Cfg_bin – this is the core of a configuration block, with extensions such as C_STRG$, C_CODE and C_WORD to return data. When the main program has finished asking you to enter details, it loads Cfg_bin and adds the config block to the end of this, then offers to save it and you have to type in a suitable filename for where to save it.
- Bsconfig_doc – the instructions

You also get source files which include basconfig_bas (uncompiled program), chbase_rext (an extension used by the program), basconfig_cfg (the program's own config block – yes, you can configure basconfig with config, just to confuse you) and cfg_asm (the assembler source for the program's basic extensions).

Figure 2 shows the program in action. I was using v1.13 of the program for this, other versions may look slightly different. The first thing you have to enter is the number of items required, so you should plan this carefully beforehand for two reasons – you have to specify the number of items in advance plus you should keep a written record of the config block details in case you need to recreate it from scratch later on since Basconfig can't edit blocks it has created.



```
Configuration Extensions Level 1
This program, supplied by QLaboratories, produces a file that
allows the use of the standard QJump Configuration program in
conjunction with Q_Liberated SuperBASIC programs.

Bug fix & LONG WORD handling by Norman Dunbar, 5th April 1994.

C_VERS$, C_NAME$ extensions by Dilwyn Jones, 2nd October 1995.

Larger BasConfig display by  Dilwyn Jones, 16th February 1999.

Enter the total number of configuration items required for
your file or press ESC to quit> 2

Enter software name> David's Program

Enter software version> 1.00

Select type for item #1> String

Type selection keystroke for the item> A

Enter maximum length of the string> 62
```

Figure 2: Basconfig display.

Next, you need to enter a version number and name for your program. Version numbers usually start at 1.00, and may be up to 8 digits long in this version. This allows the 3-part version numbers along the lines of main_version.sub_version.release_number e.g. 12.34.56 The details are of course up to you. The program name can be up to 25 characters long and should usually be a meaningful name such as the name under which you distribute your program.

Next, for each item, it asks which data type you require. It starts with String type, but by pressing the space bar you can change it to other options:

String – strings, which can include filenames. Some versions allow strings to be explicitly specified as a normal string, a filename, a directory, or a filename extension. Not all versions of Config can correctly and fully handle all of these.

Char – character, a single character or keypress.

Code – a number and an associated string, such as 0=No and 1=Yes

Select – this type is not supported in this version.

Byte – a byte (8 bit, 0–255) value

Word – a 16 bit unsigned value from 0 to 65535

Long – a 32 bit long word value from 0 to 2.147483e9 or 2^31.

Once you have selected the type, you have to assign a selection key. While this is ignored by some versions of Config and Menu Config, you have to specify it because some configuration programs list all the options preceded by a selection key, so it is important to specify one to ensure each item can be uniquely selected, but it is not the end of the world if you duplicate selection keys. I usually assign A to the first one, B to the second and so on, but this is of course up to you. Next you are asked for item range in most cases, e.g. if the item is a string, it asks for the maximum length. A good example here is filenames – QDOS and SMSQ allow these to be up to 36 characters long, or up to 41 characters if you include drive names, so for filenames you might enter 41 (or 42 in fact, as it is usually better to make strings an even number length). For characters, it asks for the default character, the value it takes until you later modify it with Config. It then asks for a description of the item – this is the prompt or question Config displays for this item when you configure the program, e.g. "Enter default filename ", without the quotes of course. Be a little bit careful about when you press Esc to finish the entry, as you need to be aware how many times to press Enter to control whether any input is on the same line as the prompt, or on a new line, or even with a blank line in between. Description or prompt entries can span several lines of text if you wish, although it is usually best to keep them as short and informative as possi-

JOCHEN MERZ SOFTWARE

Kaiser-Wilh.-Str. 302 D-47169 Duisburg
http://SMSQ.J-M-S.com SMSQ@J-M-S.com

NEW! QMENU Version 8! NEW!

It has taken a long time ... but here it is: **QMENU Version 8** and **The Menu Extension Version 8**
Most Pointer Environment users already know it: the Menu Extension. It is an interface which provides ready-made menus like file-selector boxes, simple-choice-menus or select from a list. QMENU is a guideline how to use it from BASIC, Machine code or maybe other programming languages which allow Machine code interfaces. It explains how to use it with various examples in BASIC and Machine code. You are allowed to use it in your own programs and you may even sell it under license. The Menu Extension also contains the Scrap Extension ("clipboard).
Multi-column menus, file-select with tree and view option, FileInfo II support - just the FileSelect menu on its own is a beatiful extension to your system.
QMENU has not been advertised for quite a while, as the last version 7 manual was not updated in the past few years, while the Menu Extension itself got updated here and there. However, many updates in the Menu Extension and several user inquiries made me think about releasing an updated version of **QMENU**. The manual has been completely revised and reflects all the minor and major changes and add-ons: from the assembler-side, from the BASIC programming side, and also from the user's side. You get a 42-page printed manual, a floppy disk with updates keys, updated help texts for QD Hyperhelp and updated and new examples.
Please note: The Menu Extension from version 7.65 onwards works only under SMSQ/E V2 (e.g. QPC2 or systems with high-colour screen drivers). If you run the "old" QL Pointer Environment, you should stick to your old Menu Extension. English only (a German version of MENU_rext is also on the disc, but no German documentation).
Some of the changes since version 7.04 (the last "officially" documented one) are:
DSEL (Directory Select) allows up to 10 devices
RSTR (Read String) has additional parameters (which force the values entered to be ints, floats, not empty, disables ESC etc.) It can also be used to enter hidden passwords.
Timeout feature has been added to RPER (Report Error) and ITSL (Item Select).
Some menus have got a MOVE facility.
New menu SYSS (System select) provides fast selection of items from the Hotkey buffer history, currently running jobs, Things in your system, Executable Things in your system). Just one call and the System Select procedure collects all the information for you and provides it in a list - very easy selection.
Hotkey buffer history now available in the file-select instead of cycling through the "previous" ones.
All this, bug fixes and more - available NOW.
To order, please send letter, fax or E-Mail or place an order through the secure order form on **SMSQ.J-M-S.com** (you will find screenshots on the website too). **Prices (all in EUR):**
If you do not own an old version of QMENU 24.90 + 5.- postage.
If you already own QMENU (Version 7 or before): 17.90 + 5.- postage.
Special offer: For QL Today subscribers only: QMENU (Update or new): 14.90 + 5.- postage.

We accept VISA, MasterCard & Diners Club online and offline! Amex only by mail or fax, not email!
New payment methods for our customers: Money transfer to "local" account in many countries!

- Deutschland: Jochen Merz, Account 493 50 431, Postbank Essen, BLZ 360 100 43
- Österreich: Jochen Merz, Account 85055317, PSK Wien, BLZ 60000
- Switzerland: Jochen Merz, Account 60-690080-4, PostFinance, Clearing-Nr. 09000
- The Netherlands: Jochen Merz, Gironummer 3258439, Postbank NL Amsterdam
- and from all other countries in EUR with IBAN and BIC to account
  Jochen Merz, Deutsche Postbank AG, IBAN: DE21 3601 0043 0611 1004 37 / BIC: PBNKDEFF 360
- UK customers can pay in £ (convert EUR prices above to £ by multiplying with 0.86) to
  Jochen Merz, Account 83795395, Citibank UK, Sort code 30-00-45
  or send cheques in £ - no fee for UK sterling cheques!
- US customers can pay in US$ (convert EUR prices above to US$
  by multiplying with 1.32) - no fee for US cheques in US$!

Cheques payable to Jochen Merz only!

Price list and offline exchange rate valid until 15th of Jan 2009 ble. Be aware, for example, that if strings can be up to 62 characters long, they should be input on a separate line, as there won't be room for a prompt and a long input on a single line, and inputs can't normally straddle 2 or more lines.

When using the Char type, you can specify ranges of characters which can be allowed, e.g. digits, upper case letters, lower case, other printable characters, and cursor characters. This allows Config to limit the characters which may be pressed, e.g. letters only.

Code is a difficult type to master until you get used to it. It lets you specify a few values such as 0 for No and 1 for Yes, but to make it easier Config displays a short text for each value, while the extensions return the code value so that your program can do a simple SELECT ON statement to determine the course of action to be followed for a particular selection. So you would enter a code number of 0, then set its string (or "description text") to No. Next, you might enter a number 1, and enter the text Yes, then possibly a number 2 and text "Don't Know" if this is intended to make the program take a third course of action if neither Yes nor No is appropriate. Each option can have its own selection letter, although not all config programs allow this. Useful, but hard to master without practice.

For byte, word, or long word values you have to enter a default value, the range of values (minimum and maximum) they can take, and the description text for that option.

Finally, the Select type offered by the QJump Config Block specification is not supported by Basconfig and cannot be used in this version at least since it can only be meaningfully used with Level 2 Config blocks.

The basic extensions take the form C_typename(number). For example, to return the first config block string, you might use LET s$ = C_STRG$(1), or to fetch the second byte value, you might use C_BYTE(2). The number refers to the number of that type of item, not the total item, e.g. if your config block has just one string, and one byte value, you would use C_STRG$(1) and C_BYTE(1) not C_STRG$(1) and C_BYTE(2).

Please note that some versions of Basconfig require you to set PROG_USE and DATA_USE commands to the name of the drive and directory containing the program and its files, so that it can find its own files to create a config block. Some versions let you configure the program to tell it where to find these files.

Some versions of Basconfig seem to have a small problem with string type items. Suppose we need to set up a string of 42 characters maximum to hold a filename. We set the default string to "FLP1_", which of course is less than the maximum length. Later, we reconfigure the config block and assign a longer string such as "WIN1_DIRECTORYNAME_" and that causes the block to become corrupted for some reason. I don't know why this happens, but I do know how to avoid it – always create the config block with the longest possible string, in this example you could use 42 full stops, for example. It seems to work OK if strings become shorter, but not if they become longer than they were originally configured, as though insufficient space is assigned during creation to hold the maximum length of the string unless that string is as long as it is allowed to be.

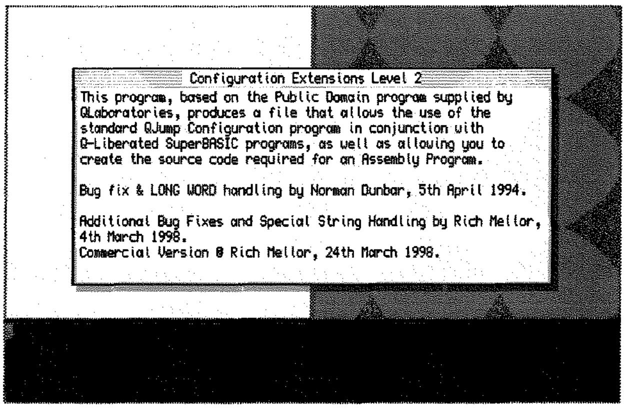
Verdict: Easy way to start writing config blocks for your programs, but only supports basic programs, not assembler or C. Cannot alter config blocks it has created itself. Suffers slightly from being updated by several authors, so you are never really sure if you have the latest version.

Q-Config

I used version 1.21, by Rich Mellor. It includes a copy of the original Basconfig, in an enhanced form with a few extra bug fixes and Special String Handling exclusive to this version of Basconfig.

Q-Config is developed from and based upon Basconfig, but now with the addition of the Select type, support for assembler output, support for level 2 config blocks and including the bug fixes included in the various earlier versions of Basconfig.

When you start the program, it initially looks like Basconfig, but with a slightly smaller display than the version 1.13 of Basconfig I was using., and with the text 'Configuration Extensions Level 2' across the top.



```
Configuration Extensions Level 2
This program, based on the Public Domain program supplied by
QLaboratories, produces a file that allows the use of the
standard QJump Configuration program in conjunction with
Q-Liberated SuperBASIC programs, as well as allowing you to
create the source code required for an Assembly Program.

Bug fix & LONG WORD handling by Norman Dunbar, 5th April 1994.

Additional Bug Fixes and Special String Handling by Rich Mellor,
4th March 1998.
Commercial Version © Rich Mellor, 24th March 1998.
```

Figure 3: Q-Config opening screen

Even if you are familiar with the original Basconfig, it pays to read the new instructions, as Rich Mellor has greatly extended this program and expanded on the instructions. It uses the same extension names as the earlier version, but now includes a new function called C_SEL$ to return the values of Select type items. For level 2 config blocks, it can handle both Registered and Unregistered names. Basically, what this means is that if the ID name is registered, it is guaranteed to be unique, so MenuConfig can store the program details in its menuconf_INF file. This allows MenuConfig to "learn" the details from an older version of a program, then you upgrade to a new version and MenuConfig can use the information from the old version to apply to the upgraded program, so that you don't need to reconfigure the new version of a program from scratch - MenuConfig can "update" a program's config block if you like.

To get a registered ID, you need to ask Jochen Merz or Wolfgang Lenerz (the SMSQ registrar) to issue you with a unique name which will not be allocated to anyone else's programs, e.g. I might get something like "DEN" or "DAV". You can see examples of registered IDs on Wolfgang Lenerz's website at

http://www.scp-paulet-lenerz.com/smsqe/Add1.html

I couldn't apply to have DDE because it is already used for Data Design, but I could probably apply for other abbreviations of my name. Having unique registered ID names helps prevent any chance of confusion between program details stored in MenuConf_inf by MenuConfig programs when configuring programs.

When you run this program, it asks first for the number of items you require in this config block, then prompts for level 1 or level 2 standards. Level 1 proceeds pretty much the same as earlier Basconfigs, except that you can choose between binary output (like the older Basconfigs, for use with basic programs) and assembler output (for programs written in assembly language).

Level 2 will ask you for additional information such as whether the ID name you intend to use to identify the configuration block is a unique and registered name, or a simple unregistered ID. It will begin with an upper case letter, and will be 3 characters long if a registered name, or 2 characters if unregistered.

You are then asked to enter an item ID type for the item in the configuration block. Here, it starts to get a bit more complex! This can be a GLOBAL or UNIQUE type. Global Items allow you to set a default in one program and whenever you configure another program which contains that same Global Item, you can update the program so that it will adopt the same setting for that default as the other program. Wolfgang Lenerz's web page currently lists 5 Global Item IDs:

_COL Main Colourway : Byte : range -1, 0 to 3.
_COS Sub-window colourway: Byte : range -1,0 to 3
_COB Button Colourway : Byte : range -1, 0 to 3.
_COE Explanation window Colourway : Byte : range -1, 0 to 3.
_FFU Flash-frequency for update icon : Byte :f 0 (steady) or ticks

This behaves like a CODE item, and returns code values of 0-3 for the usual 4 colourways, and 255 where the value -1 is inferred.

A Unique item ID type is built up as the 3 character Program ID Name plus a number from 1 to 9, or letters if more than 9 items. So if I'd been given the unique ID DEN, my Unique config block items could be DEN1, DEN2, DEN3 and so on.

From there on, it is pretty much like the original Basconfig, except for the Select data type, which can only really be used with Level 2 MenuConfig. Select data type allows you to present the user with various options and the user can select one or more of these options to be effective. If you use level 1 Config program, only one of these options can be selected. You need to read the instructions carefully to correctly set up and use this data type and its associated C_SEL$ extension.

For assembler users, the instructions also show how to use the item pre-processing and post-processing routines, which was useful because I have never seen examples of how to use this facility, which is normally set to zero to indicate no routines are included.

Verdict: A great update to Basconfig which retains the ease of use of the original Level 1 Basconfig but now supports Level 2 as well, plus provides other features not in the original. Also cannot alter config blocks it has created itself.

[Editor's note: We never thought about "personal" IDs for personal usage. Mind you, the number of QLers should easily allow to fit into three-character-abbreviations. The registration of IDs is done by Wolfgang Lenerz only (not Jochen Merz anymore), so if you would like to get "your" personal ID, please contact Wolfgang - but maybe not all at once!]

Helping Quanta Out

by Geoff Wicks

Quanta has not responded to QL Today's offer of 2,000 words to present their case in QL Today so we have decided to help them out:

"Members of Quanta will have noted a much improved print quality in the Quanta Magazine."
QL Today V10 I4 P6

"Quanta can proudly boast that it is the only QL publication that has survived as long as the QL, but for almost 10 years now it has been struggling, and has become a poor shadow of what it should be. In the last few months the Quanta committee have put a lot of effort into restoring its reliability and readability, and are having some measure of success. We at QL Today wish them well."
QL Today V10 I5 P3

"Roger is a QL-er who lives in Spain and he is slowly modifying the Psion suite for use on high resolution GD2 colour screens. He sees this as being very much a Quanta project and his work is a major acquisition for the Quanta library."
QL Today V10 I5 P28

"The last item of the Manchester weekend was the Quanta AGM. This has been a good year for Quanta and the chairman was able to make a positive report to members."
QL Today V10 I5 P29

"The committee have also been looking at shows and hope to stimulate interest in these by paying for bed and breakfast costs for 2 or 3 lecturers and demonstrators."
QL Today V10 I5 P29

"Roger Godley clearly sees his work as being a Quanta project, and thus (the programs) are unlikely to be available via the internet or other software libraries. This may disappoint some QL-ers who are not Quanta members, but then a year's subscription to Quanta would be a small price to pay for software improvements of this quality."
QL Today V10 I5 P44

"Talks and demonstrations are back in fashion at (Quanta) shows and organisers are now expected to beg, borrow or hire a computer projector."
QL Today V11 I1 P17

"Statistics may give a picture of an ailing organisation, but Quanta has made some progress in recent months. The Quanta Magazine has improved considerably; there has been a welcome improvement in publicising committee decisions; and a greater willingness to exploit some of its capital."
QL Today V11 I2 P53

"Quanta has published the first electronic edition of the Quanta Magazine available to all members. It is a bumper 40 page issue weighing in at a slim 337Kb."
"Hero of the piece is acting editor John Gilpin. He temporarily took over the editorship in 2005 even though he had no previous experience of magazine publishing. John quickly learnt to master Page Plus and, during his tenure, the magazine has vastly improved in both content and print quality. Praise also has to be given to Quanta chairman, John Mason, for the high priority he has given to the magazine, which previously had been neglected by successive committees."
QL Today V11 I4 P4

"For years Quanta members have moaned about the poor quality of the magazine, but in the last 18 months John Gilpin and John Mason have achieved something quite remarkable. They have turned the Quanta Magazine back into a serious QL publication."
QL Today V11 I4 P62

"This year I have said on many an occasion that real gains are being made in Quanta; that Quanta is lucky to have an efficient secretary; that through his hard work another committee member has made significant improvements to Quanta; that Quanta appears now more willing to exploit its capital than previously; and that Quanta has been unfairly maligned by many people over its supposed failure to fund Goldfire"
QL Today V11 I4 P63

"The Quanta Magazine has had a good 2006 and can now claim to be a serious QL publication once again. Other good Quanta news is a completely renewed website."
QL Today V11 I5 P3

"Quanta has financed by means of a loan the manufacture of 200 new keyboard membranes"
QL Today V11 I5 P5

"The greatest interest was in Dan Abbott's demonstration of the new Quanta website. Dan has little experience of the QL community because his interest in QL architecture comes via the OPD. He joined Quanta and accepted nomination to the committee to become webmaster. At the moment he is on a steep learning curve to familiarise himself with the people and traditions of the QL community."
"Dan impressed most present by his willingness to listen to members' opinions and ideas. He clearly takes a flexible view of the website he is creating, and appears to be prepared to modify it in the light of experience and possible future developments."
QL Today V11 I5 P28

"(Geoff Wicks) proposed a vote of thanks for the high quality of the minutes, but strangely no other person backed this up. Why do Quanta members find it so difficult to thank their committee when they are doing something well?"
QL Today V11 I5 P29

"In your reporter's opinion this was probably one of the best Quanta AGM weekends for some time. Both Dan Abbott's website presentation and the AGM itself had a high level of member participation and this is something Quanta should encourage."
QL Today V11 I5 P30

"Quanta now has an interesting committee. Although it is too small in numbers, it is filled with hard workers and probably for the first time in years has no dead wood."
QL Today V11 I5 P30

"Quanta's committee are planning a party to celebrate the QL's quarter centenary in 2009. It will be a good opportunity for the rest of us to show our appreciation of their work and achievements. What better way to do this than for others to take over and continue to build on their work with the same enthusiasm?"
QL Today V12 I1 P3

"The easiest thing for the Quanta committee would have been to have scrapped the Midlands venue and move the show elsewhere, possibly Manchester. However they did not choose the easy option as a matter of principle. Some years ago I had researched the demography of Quanta and had identified the Midlands as being a black spot where there was a concentration of members, but no workshops. The Quanta committee had decided to hold a show in this area as a matter of policy and they were not going to go back on this decision. They would find an alternative venue."
QL Today V12 I1 P50

"Dilwyn demonstrated the QL On A Stick to an interested audience at the Quanta Birmingham Workshop early in October. Present at the demonstration was Quanta chairman, John Mason, in case there was still a role for Quanta to play in the future development of the project."
QL Today V12 I2 P4

"I also make an analysis of the Quanta Magazine each year and the differences between volume 23 and volume 24 are striking. The average size of the magazine has increased from 34 to 41 pages and editorial content from 25 to 31.5 pages. The number of contributors has risen from 9 to 14. Well done Quanta committee!"
QL Today V12 I4 P3

"Rich's most successful lines are Spectrum and QL keyboard membranes, and the last batch of these was financed by a loan of £2,150 from Quanta."
QL Today V12 I4 P8

"NEMQLUG, who run the show, have mastered the art of running near traderless shows and, to their credit, this year they achieved the best attendance at a Quanta AGM for some time."
QL Today V12 I4 P30

"The meeting can probably best be described as short and sweet. The chairman was able to give a positive report of 2007 with successful shows held in Hove and Solihull; standards being maintained in the magazine and electronic publication proving worthwhile; a renewed website; and new committee members.
The treasurer was also able to give a positive report with the organisation breaking even."
QL Today V12 I4 P33

"At the Solihull show last year chairman, John Mason, told QL Today he could sense a renewed interest in Quanta and events since then confirm this."
QL Today V12 I4 P36

"A member of the committee told QL Today the word 'change' was frequently used in their first meeting, which immediately followed the AGM. This indicates the committee are ready for the changes ahead, but will they get more active support from the members than they are presently getting? UK QL-ers may be more dependent on the survival of Quanta than they realise."
QL Today V12 I4 P36

"Rich Mellor's most popular line is keyboard membranes, a good opportunity to remind everyone that these were financed by a loan from Quanta."
QL Today V13 I1 P55

"Logically as trading diminishes Quanta will become increasingly important for the survival of the UK scene."
QL Today V13 I1 P55

And finally:

SHUM MISHTAKE SHURELY

"Geoff Wicks, once more, somewhat mischievously seeks to diminish Quanta completely oblivious that in so doing he diminishes himself."
Quanta chairman, Quanta Magazine V25 I2 P5

"(Geoff Wicks added) that he had always felt that John (Mason) was an excellent chairman."
Quanta secretary, Quanta Magazine V25 I2 P8

Memo to Quanta: If two Quanta officers wish to mislead Quanta members about QL Today's editor they should first talk to one another to ensure they tell the same story.

The QL and Binary Music

by Stephen Poole

In QL Today of sept-october 2003, on page 36, there is a short program for playing simple music on the QL. (The 'Tempo' procedure has been rewritten for Bruno Coativy, to make it correspond exactly with actual tempo rates, as at the time I wrote the program I did not have exact data to do this). If anybody requires the latest QL_Tunes program, then I will let them have it if they send me a floppy-disk, (without S.A.E if outside the Euro-zone), via the editor.

[Editor's note: I suggest we put it onto the QL Today website as soon as Steve lets us have it in the latest version, and we can send it out on floppy disk provided you send us 3 International Reply coupons to cover for the postage and the floppy disk ... no need to send empty disks around].

My first attempts at writing music were full of confusion, as I imagined that the QL BEEP parameters should correspond to some sort of scale, as in MIDI for example. I had bought books on computer music, and noticed that MIDI values corresponded to decimal-wave values. But DECIMAL?!? I had always believed that, (since Pythagoras), harmonics were combinations of WHOLE numbers! (1/2, 2/3, etc., Tierce, Quint seventh etc.). Just take a look at the following MIDI note-values, taken at random:

No.	*Name*	*Octave*	*Value(Hz)*
24	doh	1	32.70
38	ray	2	73.42
52	mi	3	164.81 etc.

This strange 'decimal'-pitch situation has existed since musical instruments were first standardised, to allow them to accord harmoniously in orchestras, if with difficulty. There is a simple method of constructing any gamut of musical notes, so I wrote a program where the QL would do the hard work, and then tried to adapt those scales to the QL.

But to no avail, as the QL BEEP tones do not appear to correspond to any logical sequence.

However, I did experiment quite a bit, and ended up by noticing an incredibly regular suite of whole-number notes.

Briefly, If you set middle 'C' (Doh) at 512Hz, and then calculate all the other notes from there, you end up with an almost complete range of notes covering the entire audible range, (20Hz - 20,000Hz), and almost all of which are WHOLE numbers and EXACT harmonics from 1/3rd right down to 1/19th....

So then I wanted to experiment the range with actual music, but current orchestral instruments

are all based on MIDI-style 'decimal' notes starting from no. 64, 'La', octave 4, at 440Hz, (one of the few MIDI whole-numbers). So all traditional instruments are made to correspond to decimal (approximate) tones, meaning they cannot play the new integer gamuts.
Not to be discouraged, I went to visit the Senior Laboratory Technician at our local High School, and he set up their Sound Oscillator Equipment based on the new (Integer) scale, and examination of their Oscilloscope output proved that playing fundamental notes also played all harmonics down to 1/19th simultaneously.

So that was proof of the system. All that remains now is to accord an electronic keyboard with the new Integer Scale and play exactly harmonic music. Why a keyboard? Because otherwise you would have to make a whole new set of orchestra instruments. After being used to 'traditional' harmonies, the new Integer ones sound too 'clean' and 'pure' to be true, but as I cannot play keyboards, I will let others do the remaining work of playing the instruments and listening to the new integer music.

One thing that strikes me is why this was apparently never done before? The calculations are perfectly accessible by hand, and reproducing the new notes would not present any particular difficulty for most instrument-makers. No doubt, somewhere in India there is at least one music Guru using this scheme, as they are not only very versatile musicians using gamuts virtually unknown to western music, but also excellent mathematicians, and so they have been for countless generations.

If any readers try playing music on a so-redefined keyboard, I would be grateful if they would let the editor have their comments.

n.b: To keep the program simple, whilst it calculates the frequencies of notes, it does not place them in each octave in ascending order. To do so would greatly complicate the coding. But with the entire range on screen for each harmonic set, I will leave the ordering operations as an exercise for you. I would ask readers to bear in mind that such sorting depends on the way in which the human brain interprets pitch, which is not directly dependant on any obvious characteristic.
As an example, the fifth octave frequencies are: doh-512, ray-576, me-648, fah-729, sol-768, lah-864, si-972.
If you examine the first octave figures, you will see that to get the correct order, you have to jump from column to column, listening to each generated pitch as you go, and deciding in which octave each note fits. If anyone knows a method of programming this I would be glad to hear of it. But strangely, in this age of all-automation, there are still some operations which can only be done by hand! If I knew the theory, I would also try tackling all other keys, including majors and minors etc. Any offers?

```
100 ::
110  REMark Binary_Gamuts_bas, by S.Poole. vtsept2006
120  REMark Notes outside audible range are noted as '0'
130  REMark Note how new decimal notes are rare:
140 :
150  OPEN#1,con_16: WINDOW 512,256,0,0: CLS: UNDER 0
160 :
170 FOR Harmonic=1 TO 19 STEP 2
180     h_2=Harmonic/2: PRINT \ '  Harmonic: ';Harmonic
190     :
200     FOR hz=5 TO 13
210         h=2^hz: UNDER 1: PRINT h,: UNDER 0
220         :
230         FOR n=1 TO 6: h=h*h_2: IF h‹20 OR h›20000: h=0: END IF : PRINT;h,
240         PRINT
250     END FOR hz: i$=INKEY$(#1,1000)
260 END FOR Harmonic: CLS: WINDOW 512,206,0,0: STOP
270 ::
```

Public Holidays

by Geoff Wicks

I once wrote a calendar program for the QL. As I write this I can picture some of our more seasoned readers stifling a yawn and muttering "So what!".

Calendar programs are old hat. They were one of the staples on which many of us cut our programming teeth. The first book on programming I bought had a calendar program that just fitted in to the 1K of a ZX81.

Some years ago David Denham wrote a series of articles on clocks and calendars for QL Today. If you are interested in looking them up you can find them in Vol. 6 issue 5 page 15; Vol. 6 issue 6 page 48; Vol. 7 issue 1 page 45; and Vol. 8 issue 4 page 16.

Stephen Poole has also written a calendar program that appeared in Vol. 9 issue 2 page 48. This is the only program I have seen that allows you to work in either the Julian or the Gregorian calendar.

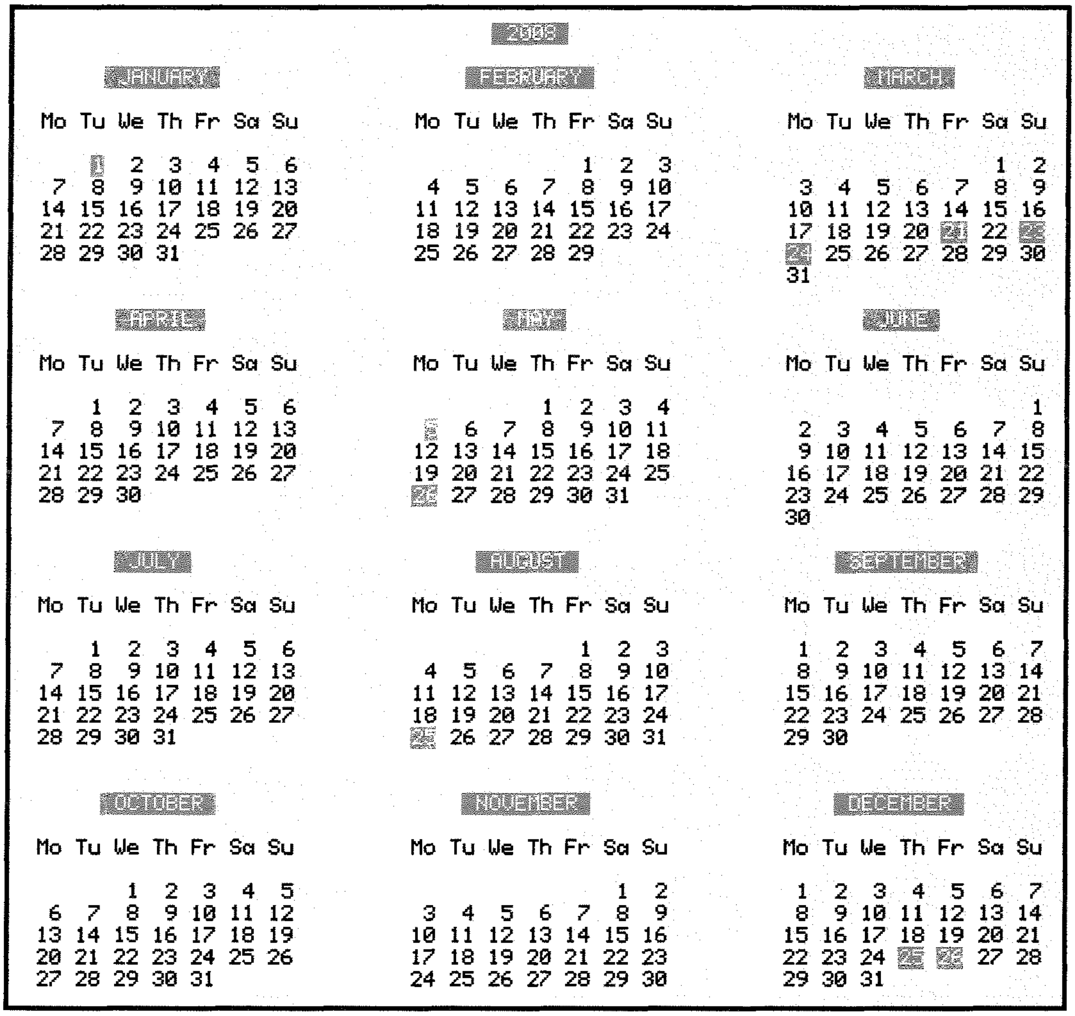
I have seen many calendar programs but there is something missing in them all but my own. Let's look at a practical example. Suppose it is your job to plan the date for the next Quanta AGM. This is a more difficult task than you might think because the constitution says that members have until 1st February to nominate candidates for office; that the members have to be given at least 28 days notice of the business of the meeting; and that the latest date for the meeting is 30th April. In theory this is a window of about 9 weekends on which the AGM can be held. In practice it is only half that because the business of the meeting has first to be edited, printed and distributed. There is a further complication that the Easter weekend falls somewhere in that window.

You can call up your QL calendar program to enable you to choose the best weekend, but you also have to hunt for your diary because your calendar program does not tell you the date of Easter.

Wouldn't it be nice if your calendar program also displayed public holidays? That is what my program does. If you want to look at it you can download it from the downloads page of my website:

http://members.lycos.co.uk/geoffwicks/justwords.htm

(Your system will have to be able to display GD2 colours and have a resolution of at least 800 x 600. This is the minimum resolution that can comfortably display a calendar for the whole year on a single screen.)



```
                                     2008

      JANUARY                  FEBRUARY                  MARCH

Mo Tu We Th Fr Sa Su     Mo Tu We Th Fr Sa Su     Mo Tu We Th Fr Sa Su

    1  2  3  4  5  6                  1  2  3                     1  2
 7  8  9 10 11 12 13      4  5  6  7  8  9 10      3  4  5  6  7  8  9
14 15 16 17 18 19 20     11 12 13 14 15 16 17     10 11 12 13 14 15 16
21 22 23 24 25 26 27     18 19 20 21 22 23 24     17 18 19 20 21 22 23
28 29 30 31              25 26 27 28 29           24 25 26 27 28 29 30
                                                  31

       APRIL                     MAY                      JUNE

Mo Tu We Th Fr Sa Su     Mo Tu We Th Fr Sa Su     Mo Tu We Th Fr Sa Su

    1  2  3  4  5  6               1  2  3  4                        1
 7  8  9 10 11 12 13      5  6  7  8  9 10 11      2  3  4  5  6  7  8
14 15 16 17 18 19 20     12 13 14 15 16 17 18      9 10 11 12 13 14 15
21 22 23 24 25 26 27     19 20 21 22 23 24 25     16 17 18 19 20 21 22
28 29 30                 26 27 28 29 30 31        23 24 25 26 27 28 29
                                                  30

       JULY                    AUGUST                  SEPTEMBER

Mo Tu We Th Fr Sa Su     Mo Tu We Th Fr Sa Su     Mo Tu We Th Fr Sa Su

    1  2  3  4  5  6                  1  2  3      1  2  3  4  5  6  7
 7  8  9 10 11 12 13      4  5  6  7  8  9 10      8  9 10 11 12 13 14
14 15 16 17 18 19 20     11 12 13 14 15 16 17     15 16 17 18 19 20 21
21 22 23 24 25 26 27     18 19 20 21 22 23 24     22 23 24 25 26 27 28
28 29 30 31              25 26 27 28 29 30 31     29 30

      OCTOBER                  NOVEMBER                 DECEMBER

Mo Tu We Th Fr Sa Su     Mo Tu We Th Fr Sa Su     Mo Tu We Th Fr Sa Su

       1  2  3  4  5                     1  2      1  2  3  4  5  6  7
 6  7  8  9 10 11 12      3  4  5  6  7  8  9      8  9 10 11 12 13 14
13 14 15 16 17 18 19     10 11 12 13 14 15 16     15 16 17 18 19 20 21
20 21 22 23 24 25 26     17 18 19 20 21 22 23     22 23 24 25 26 27 28
27 28 29 30 31           24 25 26 27 28 29 30     29 30 31
```

Why am I the only person who has included public holidays in his calendar program? There is a clue in the program description on the website. It only displays English public holidays. And when I say English I mean just that. The program does not correctly display Scottish or Northern Ireland public holidays.

In other words it is easy to write a universal calendar program, but it is impossible to write a universal public holiday program. Each land has its own public holidays and even within a land there can be regional differences.

In this situation QL-ers have an advantage over users of other computer systems. The ease of programming in SuperBasic means that it is possible to write a tailor made calendar program. Even if you have no wish to write your own program, you could still amend someone else's to include your country's public holidays. Nor do you need to stick to public holidays. If you are Muslim, Jewish or other religion you may be able to program in your holy and feast days. If you wanted you could even tailor your program to show family birthdays and anniversaries.

Public holidays come in three types. There are fixed days; occasionally variable days; and totally variable days.

FIXED DAY PUBLIC HOLIDAYS are the easiest to program. We all know that Christmas Day is on 25th December and News Years Day on 1st January. In your print routine you will have commands like `PAPER 7 : INK 0`. A simple line of basic reverses these:

```
IF month = 12 AND day = 25 : PAPER 0 : INK 7
```

(But do not forget to reset to `PAPER 7 : INK 0`)

OCCASIONALLY VARIABLE PUBLIC HOLIDAYS can come in various sorts, some of which are easier to program than others. In the Netherlands the monarch's birthday is an important public holiday. If a monarch dies or abdicates you will have to reprogram your basic. (As it happens the present Queen's birthday is at the end of January. As this is not a suitable time for the street markets and parties that are a feature of the day the celebration still takes place on her mother's birthday.)

Many decades ago the UK elected a Labour government after many years of Conservative rule. They rewarded us by an extra public holiday and what better day than 1st May - the day of the worker.

This was simple to program:

```
IF month = 5 AND day = 1 : PAPER 0 : INK 7
```

However the Labour government was short-lived and when the Conservatives came back in power the idea of a public holiday to celebrate the worker was abhorrent. But no politician would dare take a holiday away, and so they cunningly changed the holiday from 1st May to the first Monday in May and in so doing gave everyone a long weekend. The programming now has become a little more difficult:

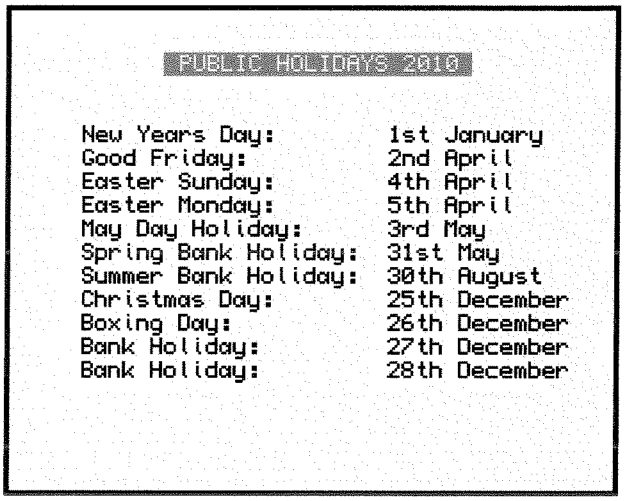
```
IF (month = 5 AND day< 8) AND cd/7 = INT(cd/7) :
PAPER 0 : INK 7
```

(In my program cd/7 = INT(cd/7) identifies the day of the week as a Monday.)

Some years later the government replaced the totally variable Whitsun holiday by an occasionally variable one of the last Monday of May. Then the code becomes:

```
IF (month = 5 and day> 24) and cd/7 = INT(cd/7)
: PAPER 0 : INK 7
```

The British have another unusual public holiday rule. If Christmas Day, Boxing Day (the day after Christmas) or New Years Day fall on either a Saturday or Sunday then an extra public holiday is awarded. Thus in 2010 Christmas Day is a Saturday and Boxing Day a Sunday, so 27th and 28th December become public holidays.



```
              PUBLIC HOLIDAYS 2010

New Years Day:           1st January
Good Friday:             2nd April
Easter Sunday:           4th April
Easter Monday:           5th April
May Day Holiday:         3rd May
Spring Bank Holiday:     31st May
Summer Bank Holiday:     30th August
Christmas Day:           25th December
Boxing Day:              26th December
Bank Holiday:            27th December
Bank Holiday:            28th December
```

A programming rule to cover that possibility is quite complicated.

As a final example of an occasionally variable public holiday I shall return to the Netherlands. There is a continual debate about whether Liberation Day on 5th May is or should be a public holiday. The official answer is that it is a public holiday once every five years. However many firms treat it as an unofficial public holiday each year. It is not easy to decide whether or not to program Liberation Day as a holiday or not.

TOTALLY VARIABLE PUBLIC HOLIDAYS are the biggest headache and they are very common in lands where the public holidays are still largely church based. The most obvious is Easter which occurs on a different day each year.

The rule is in fact very simple. Easter Sunday is the first Sunday after the first full moon on or after the Spring Equinox. To simplify matters the church cheated by defining the Spring Equinox as 21st March although it is in practice variable.

The big challenge is how to turn this into a computer program and, frankly, I suspect the only QL-er who would have had both the astronomical and programming skills to do this would be Freddy Vachha. If it is any comfort to our collective bruised ego then the algorithm the church uses to calculate Easter is not strictly accurate and is only guaranteed to work from 1700 to 2299.

Fortunately others have done the work for us. In the Quanta library there are two programs for calculating Easter. They can be found in the folder "ONEFILEPROGS" on disk UG02. The first program Easter_bas is by Darren Jones and the second program Easter2_bas by John Tanner. Even if you are not interested in Easter programs, it is still instructive to look at these as John Tanner has reworked Darren Jones' program to make the basic more compact.

If you are not a member of Quanta then you will also find the Easter algorithm in David Denham's third article (Vol. 7 issue 1 page 45). Finally the website:

www.bbc.co.uk/dna/h2g2/A653267

gives the algorithm in a suitable form for use in a basic program.

Once you have worked out a date for Easter the other holy days are easy to calculate. Ascension day is the Thursday after the 5th Sunday (i.e. 39 days) after Easter and Whit Sunday is 7 weeks or 49 days after Easter.

If you want to explore the possibilities of QL calendar programs I have a final word of warning. Many of the published programs stem from the early years of home computing and are not written to the standards of the present day. In the early years memory was small and a good program was one in which the code was as compact as possible. Thus in John Tanner's Easter program you will find a line:

```
a = j MOD 19 : b = j MOD 4 : c = j MOD 7 : t=19*a+24
```

Nowhere are you told what a,b,c,j and t are. Today with our abundance of memory we would use "year" as the variable instead of "j" and be told in a REM statement that b = j MOD 4 is a leap year calculation.

The best place to begin is David Denham's articles. They were written about 6 years ago and are much easier to follow.

Adding public holidays to a calendar program can be an interesting challenge, but one in which you could well be on your own. If you are successful then do not be selfish. Share your program with your fellow countrymen.

Letter-Box

George Gwilt writes:

Letter Commenting on Norman Dunbar's replies to my comments on part 21

In Letter-Box of QL Today Volume 13 Issue 1 Norman Dunbar thinks that "the CLOSE% command can only close SuperBasic channels (or those contained within a compiled SuperBasic program)".

The syntax given in TK3 is:

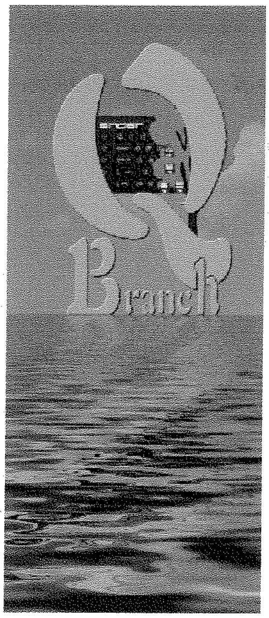
```
CLOSE% channel      close an internal channel
```

The word "internal" is used to distinguish this from a SuperBASIC channel.

It is interesting to speculate how CLOSE% might be programmed to do this.

The number of an internal channel is its position in the system channel table (0, 1 2 etc.). The address of the base of the system channel table is $78 bytes from the start of the system variables. We can thus find the entry for the channel to be closed. This is a long word pointing to the channel block for that channel if the channel is open. If the channel is closed the long word pointer is negative. Simply to close that channel, if it is open, is quite easy. Just use IO_CLOSE with the ID of the channel in A0 and it's done. The ID of the channel is a long word with the channel number in the bottom word and the tag, which is in the word $10 from the start of the channel block, in the top word.

However, to implement Norman's suggestion we must test whether the channel belongs to an S*BASIC job, which means either the master BASIC or a daughter BASIC. This can be done, in SMSQE, since all S*BASIC jobs have "SBAS" in the long word just before A6. We thus have to test this for a target channel. The ID of the job owning the channel is in the long word $08 from the start of the channel block. We can use this to find the address of the job control block of the



Feeling out on a Limb?
Reach out for QBranch
Suppliers of Quality QDOS/SMSQ products
Hardware and Software

20 Locks Hill
Portslade
Sussex
BN41 2LB
UK
Mobile: +44(0)7836 745501
email : sales@qbranch.demon.co.uk
www.qbranch.demon.co.uk

We can accept payment by Visa, Mastercard, Maestro and Switch. We also accept sterling cheques drawn on a UK bank. or payment direct to our bank:
Natwest
Sort code 60-17-01
Acc. 84534826
BIC : NWBKGB 2L
IBAN :
GB34 NWBK 6017 0184 5348 26

QBranch is closing down!

After 12 years selling QL Software and hardware we have finally come to the end of the road. We would like to extend a big thank you to all the customers who have supported us over the years and we hope that you will all continue to use the QL and its emulations.

We don't hold much stock any more but see below for closing down sale items.

Text87 plus 4	£25.00
Fountext94	£ 5.00
Qliberator	£15.00
Basic Linker	£ 5.00
WinEd	£ 5.00
QBasic	£ 5.00
HyperHelp for Basic	£ 5.00

Games

Super Games Pack	£ 8.00
Arcanoid	£ 2.00
Mine Field	£ 3.00
Pipes	£ 3.00
FireBirds	£ 2.00
QShang	£ 2.00
Black Knight	£10.00

For most of the items above I have only one or two copies. I also have most back issues of QL Today @ £1.00 each. Prices do not include Post and Packing.

Any items left after Jan 31st will be disposed of.

job owning the channel. Then we find the value of A6 from the long word $58 bytes from the start of the job control block. So we could, with a bit of programming, avoid closing channels which are not S*BASIC.

What of compiled tasks? I can only speak for those compiled by Turbo. I have a program ADJ_DS which will report, and allow alteration of, the dataspace of all jobs and the stack space of Turbo compiled jobs. Thus I must be able to pick out these Turbo compiled jobs. The code I have used could, obviously, be used to allow the closing of channels owned by such jobs. In fact I find these jobs by checking that the first few bytes of the program contain a Turbo version number.

Well, it is theoretically possible that CLOSE% could do something like that. In fact it does not. It does just close the channel with system number "channel". What is slightly amusing is that, while most accesses to channels require the whole ID, CLOSE% just asks for the channel number without the tag. This means that you could sometimes close the wrong channel. How? Suppose you establish which channel number you want to close. This takes some time. Even using CHANNELS from TK3, which quickly prints all system channels, takes a few seconds while you tap away at the keys. By the time you get round to typing CLOSE% 126 (or whatever), that channel may have been closed and another one opened at the same position in the channel table but with a different tag. After all the QL is a nicely multi-tasking computer.

Finally I should report that recently, while trying to find the error in a colleague's assembler program, I noticed that I could not use the floppy drive. All access was denied. I looked at the channels opened for that job and found one open to a file on a floppy disk. Since we had reloaded the drive with a second disk the SMSQE software refused to deal with it. I solved the problem by using CLOSE% to close that channel. After that the floppy disk drive was operational.

George Gwilt also writes:

Letter Commenting on a "quick reply from Jochen"

At the end of my article about resizing windows in QL Today Volume 13 issue 1, Jochen gave an explanation of why QD moved horizontally on a resize. In fact this caused me to examine resizing even more closely. I found that in some cases windows which I thought would have a stable bottom right corner in fact moved slightly, always to the left by a couple of pixels or so.

I discovered that, to maintain a rock steady bottom right, you have to do three things.

1. Find the new position for WM.PRPOS as explained in my article.
2. Make sure that the number of pixels change horizontally is divisible by four.
3. Make sure that the minimum x-size for the window is divisible by four.

I have experimented with my windows constraining the amount of horizontal change to multiples of some chosen number and found that, if the three rules above are obeyed there is no movement of the window on resizing.

So I still don't understand the quirky QD movement!

[Jochen replies: Thanks, George ... after all these years, I am not sure that the resize routine checks your points 2 and 3. They should, because you are right - it could lead to small moves by a couple of pixels to ensure the window/colours matches stipple positions, I think. I will look into QD to see if that's the case after the magazine has been finished and done, and after I finish a few small changes on QMENU].

Fun with Fonts – Part 1

by Dilwyn Jones

The QL character set is made up of a collection of character definitions gathered together in a piece of data called a Font. In Sinclair QL terminology, it should more correctly be called a Fount, but since everyone seems to use the word Font, so shall we.

Each character is defined on a 5x9 grid of dots, called pixels (which in turn stands for Picture Elements). When a character is to be drawn on the screen, it is done so using this 5x9 grid of pixels, and spaced by one pixel between adjacent characters both across and down, in effect a 6x10 pixel grid per character, but only 5x9 of these pixels can be defined for each character in the font. This 5x9 grid forms a pattern which tells the computer how to draw each character on the screen.

Please see figure 1 for a diagram showing how this works in practice. This shows how an upper case letter A is defined.

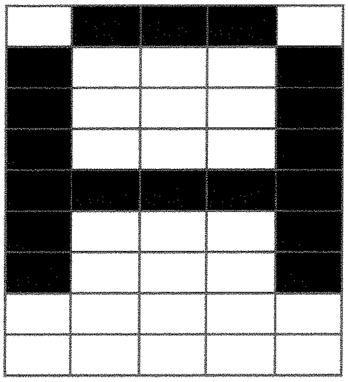


Figure 1 - Character definition 5x9 grid

In this diagram, each black square represents a pixel which is set in INK colour on the screen for this character and each blank (or white) square represents an unset pixel on the screen which is normally printed in paper colour (assuming OVER 0 is set).

This grid in effect represents a binary version of the character. Each line across is encoded into a byte value, using bits 6 to 2 inclusive, which leaves one blank bit to the left and two to the right. Since each character has 9 lines of pixels across, 9 bytes are needed to store up the dot pattern for each character. The top row of a character on the screen is stored in the first byte, and so on, down to the bottom row.

Here's a sample of what a solid character (all dots set) might look like in binary:

```
    01111100
    01111100
    01111100
    01111100
    01111100
    01111100
    01111100
    01111100
    01111100
```

Built-In Fonts

The QL has a default character set built into the ROM. This covers all of the characters which the QL can print, but split into two fonts. The first covers all characters with codes up to and including 127. The second font covers all of the characters which are referred to as "the extended character set" or those characters such as accented letters. What happens is that if you ask to print a certain character code, the QL looks to see if it is a character definition stored in the first font. If not, it then tries to see if it is a character stored in the second font. If not found in either, a stored default character is printed instead. On a standard Sinclair QL, this would be a "chequerboard" character - to see it try PRINT CHR$(31).

New Fonts

It is possible to provide the QL with a new font if you so wish. In fact, any channel of any program can in theory at least have its own character set. Over the years, many people have designed their own character set and many are available free from PD libraries and QL-related websites. There is a page on my website devoted to QL fonts - point your browser at

http://www.dilwyn.uk6.net/fonts/index.html

and look for the general QL system fonts you can download.

Font files usually have the following file extensions:

`_fnt` or	
`_font`	A standard QL font
`_fat`	A "fat" font (8 pixels wide - see below)
`_chs`	Also a standard font, chs stands for CHaracter Set

To load and use these new fonts, you have to load them into memory and use a Toolkit 2 extension called CHAR_USE to tell the system to use that font for the channels you indicate in the command.

The first thing we need to do is to find the length of the font. Let us assume it is called sample_fnt:

```
100 filename$='flp1_sample_fnt'
110 LET font_size = FLEN(\filename$)
120 base = ALCHP(font_size)
130 LBYTES filename$,base
140 CHAR_USE #1,base,0
150 FOR a = 32 TO 191 : PRINT #1,CHR$(a);
```

Line 110 checks the font size. Line 120 allocates enough space in the common heap for the font file. Line 130 loads the font into memory. Line 140 specifies that the font applies to screen channel #1, and the first font is the one replaced, while the second one remains as the font built into the QL ROM. Finally, line 150 prints all of the characters so that you can see what they look like and which have changed.

The CHAR_USE command takes the following form.

```
CHAR_USE [#channel,] font1_address,
 font2_address
```

If you omit the channel number, it will default to channel #1. If either font1_address or font2_address is given as a value of zero, that means "use the font in the ROM". So, to reset to the font in the ROM, you could use

```
CHAR_USE #1,0,0
```

which resets both fonts. You would also need to release any common heap space you have used with an RECHP command.

SMSQ/E (SBASIC) users have an extra option in the CHAR_USE command. If a value of -1 is given in place of 0, it means leave this font unchanged (use the existing definition). Therefore,

```
CHAR_USE #1,font_address,-1
```

means use the font definition at "font_address" for the first font, but don't change the second font.

Changing Default Fonts

You may have noticed that the CHAR_USE command only lets you change the fonts for a single channel of a single program. On traditional Sinclair QLs there is no easy way to change the default system font as it is built into the ROM, fixed and unalterable. But, on SMSQ/E systems, SBASIC has a command called CHAR_DEF which can change either or both of the system fonts if you wish:

```
CHAR_DEF font1_address, font2_address
```

As with CHAR_USE, we can supply the address of a suitable font, or we can use a value of 0 to reset to SMSQ/E's built in system font, or use a value of -1 to stick to the existing definition, unchanged.

CHAR_DEF changes the fonts used by any newly loaded programs. But programs which are already in memory may continue to use the same font as before. It also has no effect on programs which use their own fonts, such as graphics programs.

Font Definition

The first two bytes in a font file are special, as they contain the code of the lowest valid character in that font, and the number of valid characters less one. This information tells us which character codes are defined in that particular font. For standard Sinclair QL fonts, the first 2 bytes of both fonts are as follows:

Font 1: 31 96
Font 2: 127 64

What this means is that for the first font (which usually covers the character set up to the copyright symbol of CHR$(127) this font starts with CHR$(31) and there are 96+1 characters (97) in the font. This means that the code of the last character is 31+96, which is 127. So the first font has definitions for characters with codes from 31 to 127.

The second font consists of characters with codes starting at 127 and containing 64+1 (65) characters, which means it covers characters with codes from 127 to 127+64, which is 191.

Fonts of this type are generally of the following file lengths: Font 1 is 875 bytes long. Font 2 is 587 bytes long.

After the first 2 bytes in each font file, each character is defined in a series of 9 bytes as defined above. So for the default font in the ROM, CHR$(31) is defined in the 9 bytes after those first 2 bytes and so on.

You may have noticed that both fonts define CHR$(127). There is a very good reason for this. The mechanism by which fonts are printed is as follows:

The system looks up the character code in the first font. If it is found within the first font, it is printed using that definition. If it is not found in the first font, the system looks in the second font to see if it exists within that. If it does, it is printed using that definition. If it doesn't exist in the second font either, the character is printed as the first valid character in the second font. To see how this works, consider the example of

```
PRINT CHR$(127)
```
and
```
PRINT CHR$(255)
```

The system decides that CHR$(127) exists in font 1, so it prints that to the screen. This is the copyright symbol.

CHR$(255) does not exist in font 1, so the system looks in font 2. It doesn't exist in that either, so the system decides it has to use the lowest valid character in the second font, which is CHR$(127), but this is not the copyright symbol. CHR$(127) in the second font is a chequerboard symbol, and that is what is printed as the default character.

Locating the Font in Memory

Although it is quite easy to install new fonts as described above, there are times when you may wish to find the location of a font in memory, e.g. when you wish to study the dot patterns for a given character. Unfortunately, neither SuperBasic, SBASIC, nor Toolkit 2 have functions which will tell us where the font is located in memory, so we have to resort to some other toolkits or extensions.

The address of a font used by a given channel is stored in the channel definition block, at an offset 42 bytes into the block (for font 1) or 46 bytes in (for font 2). What we need is an extension to return this information.
The easiest way is to raid Simon Goodwin's DIY Toolkit, and use the CHAN_L function in Volume C. DIY Toolkit is available from my website's Toolkits page at
www.dilwyn.uk6.net/tk/index.html

Once we have this extension resident, we can
`PRINT CHAN_L(#0,42)` and
`PRINT CHAN_L(#0,46)`
to see where the two fonts occur in memory.

The CHAN_L function returns a long word from the channel definition block from a given offset within the block. The corresponding extensions for returning word and byte values are CHAN_W% and CHAN_B% respectively. These channel functions take care of working out where to look for the information, since the channel definition block may not stay at a given fixed address at any given time, so the address to look at is not too straightforward to calculate.
If you have Norman Dunbar's DJToolkit, there is a function called `WHERE_FONTS` to locate the fonts:
`WHERE_FONTS(#channel,1_or_2)`
which returns the address of the font used for a given channel. 1_or_2 should be replaced with the number 1 or 2 depending on which of the two fonts you wish to locate. DJToolkit is available from the same website.

How NOT to locate Fonts!

A common problem with older QL programs is code which locates fonts incorrectly. Very old programs used a couple of PEEK_L commands, like this:

```
LET font_addr = PEEK_L(167722)
```

This is doomed to failure on modern systems as it peeks directly into the address where a channel definition block might occur on an original, unexpanded QL, and makes no allowance for the slightly different channel definition block in modern systems with pointer environment, for example.
Another method seeks to work around this to some extent, but is also doomed to failure on many systems:

```
LET font_addr = PEEK_L(42+ (PEEK_L (PEEK_L
(163960)+4)))
```

You may find this second method works on some systems, as the PEEK_L(163960) looks for the address of the base of the channel table, but it could not hope to work on systems where the system variables have moved, for example. Such systems might include a Minerva system using the second screen, or an SMSQ/E system. So it is best to use CHAN_L or an equivalent function which enables us to locate the information "legally".

Assembler

For those who write in assembler, fonts are handled using the sd.fount trap, which is trap #3 with d0=$25 or decimal 37. This is called with the following register parameters to set or reset the font:

```
d3.w = timeout
a0   = channel ID
a1   = base address of first fount (or
       zero to reset)
a2   = base address of second fount (or
       zero to reset)
```

d0 can return -1 for 'not complete' (e.g. timed out) or -6 for 'channel not open'. The following registers are preserved: d2.L, d3.L, a0 and a2.

Fat Fonts

I stated above that QL fonts are based on a 5x9 pixel grid. There is one largely undocumented special case which is an exception to this rule. If you use CSIZE 1,0 this allows fonts designed on an 8x9 grid to be used. For each of the 9 bytes used to define such a character, all 8 bits can be defined and some font editors allow you to do this. Such fonts can only be used in character width 1 (which is normally the same as character width 0 but with extra pixel spacing). Such fonts are usually referred to as "fat" fonts, or 8 pixel wide fonts, because they are 2 or 3 pixels wider than the standard characters. Sadly, these fonts cannot be used in any of the three other character widths. But although this is a largely undocumented feature, this does work across all QDOS and SMSQ/E systems I know of as it has been recognised and reproduced in all versions of the operating system to date.
In practice, if you use character width 0, 6 of the 8 pixels are displayed. This is because although the characters are normally defined using bits 6 to 2 across each byte of a definition, the first (usually blank) pixel is used for the spacing and some versions of the operating system will faithfully draw bits 7 through to 2. One example of use of such a font could be for continuously joined up handwriting fonts, for example.

Character Spacing

When we print characters on the screen, each character is usually spaced by at least one pixel (more than that in character width 1 and 3). QDOS and SMSQ/E do, however, support different character spacings by allowing us to specify different pixel increments for each character. So, while CSIZE 0,0 characters are spaced 6 pixels apart, CSIZE 1,0 characters are spaced 8 pixels apart. CSIZE 2,0 and 3,0 are obviously spaced twice as far apart.

Toolkit 2 uses the CHAR_INC command to set different spacings for text characters. Each channel can have its own spacing:

```
CHAR_INC #channel,x_increment,y_increment
```

If you do not specify a channel number, it assumes channel #1. The x_increment is the number of pixels from the start of one character across to the start of the next character and y_increment is the number of pixels from the top of one character to the start of the next character down. Normally, there would be no real use for this command apart from creating unsightly overlapping text. Try

```
CHAR_INC #2,5,10
```

to see what happens when editing a SuperBASIC program, for example. Note that the reduction from 6 to 5 pixels across causes the rightmost pixel to be lost, since printing normally begins at bit 7 of a character definition, so you get the single pixel spacing, and then the first 4 pixels of the character. After trying the messy editing, now enter

```
OVER #2,1
```

to prevent the spacing pixel overwriting the rightmost part of the previous character and see that although hardly pretty, editing is now much easier!

Note that if you change CSIZE, the increments are cancelled and reset to normal spacing for the selected character size.

Where CHAR_INC can come in very useful is for when you define smaller characters than usual.

The Toolkit 2 manual cites the following example:

"*...if there is a 3x6 character fount in a file called 'f3x6' (length 875 bytes), then a 127 column by 36 row screen can be set up:*

```
MODE 4
WINDOW 512-2,256-3,0,0  : REMark clear of edges of screen
CSIZE 0,0               : REMark spacing of 6x10
CHAR_INC 4,7            : REMark spacing 4x7
fount = ALCHP(875)      : REMark reserve space for fount
LBYTES f3x6, fount      : REMark load fount CHAR_USE fount, 0: REMark single fount only"
```

Using non-standard spacings like this also carries a warning in the Toolkit 2 manual:

"*If the increment is set to less than the current character size (set by CSIZE) then extreme caution is required as it will be possible for the display driver to write characters (at the right hand side or bottom of the window) partly outside the window. The windows should not come closer to the bottom or right and edges of the screen than the amount by which the increment specified is smaller than the character spacing set by CSIZE.*"

That was the reason for the "512-2" and "256-3" in the example above.

For the technically minded, the character increments are stored as 16-bit values in the screen driver data block definition at offsets $26 (decimal 38) for the x increment, and $28 (decimal 40) for the y increment. If you are using Simon Goodwin's chans_code extension from his DIY Toolkit, you can use the CHAN_W% function to read the existing spacing settings for any channel:

```
PRINT CHAN_W%(#0,38) : REMark print
  horizontal (x) increment for #0
PRINT CHAN_W%(#2,40) : REMark print
  vertical (y) increment for #2
```

If you decide to create such small fonts to use with CHAR_INC, experiment a little to see how best to define each character. It would normally be best to design characters into the top left corner of a character definition, but allowing for one pixel spacing at the left unless you specifically want the characters to join up (e.g. handwriting, or continuous box characters like an IBM character set on some printers).

In the next part, I will show you how to edit fonts, how to print with shadows, how to outline text and other things.

Programming in Assembler - Part 22
WMAN, The Window Manager

by Norman Dunbar

Introduction

Did anyone notice I wasn't 'here' last time? Didn't think so! I was in Castelmola, Sicily again for two weeks enjoying the shade while my wife enjoyed the 40 degree heat and almost continual sunshine. Mount Etna was in good flow and we had spectacular views of the mountain during the day and the red hot lava flows at night. The food was splendid - as Italian food is - and we had a great time.

At the end of the last article in this series, I mentioned that we would be delving into the WMAN system next. Well, here we are. However, before we get down and dirty in the code, I need to make sure you all know what I'm talking about, so let's start with a brief introduction/reminder to WMAN and all it's constituent parts.

WMAN

Until now, we have been playing with the PE or Pointer Environment routines. These allow for a window to be outlined, the pointer to be drawn and read and so on. However, to use these few routines to write applications with multiple windows and so on, loose items, menus whatever, would be quite difficult. This isn't to say that it cannot be done, it's just difficult.

What we really need is a utility to allow us the ability to define our window structure, the loose items and so on contained within it and convert that into what QDOSMSQ really needs to have to be able to give us all the goodies we get from the PE, well, WMAN is just that.

Using WMAN we can define a window and all its contents, then use the vectors from WMAN to set up, display, remove and interact with our application without having to write code to handle everything ourselves.

George Gwilt mentioned in a comment about part 20 of this series that I treated the call to IOP_PINF as a method of finding out whether or not the Pointer Environment had been loaded. While it does indeed do this, it also returns a vector to the current location of the WMAN utilities in memory in A1.L - and it is these vectors we will be exploring in the coming articles.

A Very Brief Overview Of WMAN

Before we go on, we need to know what all the bits of the PE actually are, so there now follows a small briefing on that very subject. I won't be spending a lot of time in the discussions so if you need further information there is a very good "Idiot's Guide To The PE" available on Dilwyn's web site at

http://www.dilwyn.uk6.net/pe/peig/pe.html

if you want to read it online or

http://www.dilwyn.uk6.net/pe/peig.zip

if you want to download it to read at your leisure.

Selection Keys

A selection key is simply the key that you press - when the pointer is over the appropriate primary window (see below) - to activate some function or feature of the program in question. It may cause an action to be carried out or simply highlight an option is a menu. Normally, the selection key is shown underlined, but this is not necessary, although it is more helpful to the user of the program if it is.

Hit and Do

When the mouse buttons are in use then a HIT is what happens when you click with the left mouse button and a DO is when you click with the right one. On the keyboard, a HIT is when you tap the spacebar and a DO is when you tap ENTER. The actions carried out when you HIT or DO may be the same or may be different - it's all down to how the programmers wrote the code.

Outline or Primary Window

I have mentioned outlines before, however, for the sake of completeness, I'm reiterating here. The outline (or primary window) is the rectangle of your screen that the program will perform all it's workings within. Any secondary windows (see below) opened by the program must be fully contained within the area bounded by the outline. Of course, some programs allow you to move their windows around the screen. This also moves the outline around and wherever the window ends up when the user has moved it, becomes the new outlined area and all secondary windows will now appear within the new location.

The biggest size that an outline can be is the maximum width and height of the screen minus the shadow width and depth.

Secondary Windows

Secondary Windows are things like QMENU's "file open" utilities and so on, pop-up messages giving you error messages and anything else that takes place within the outline or primary window.

Information Sub Windows

These are small areas of the primary or secondary windows that show static text or little images or whatever. The most commonly seen and recognisable ones are those green and white stippled 'caption bars' that most PE programs have at the top of every window.

Indeed, the caption bar for most PE programs that I know of is set up with a green and white stippled information window all the way across the top of the window, then on top of that there is another plain white information window nicely centralised horizontally on top of the first one. The program name or caption is then inserted as an Information Object (see below) into this second information window.

Information Objects

Once an information sub window has been created you need something to put in it - for information purposes. To this end you need to create information objects. These can be text or blobs, sprites or patterns (see below). The most noticeable ones are the program name shown in the 'caption bar' of most PE programs.

Loose Items

Loose items are small 'buttons' with text or graphics on them. They usually have a border that magically appears when the pointer is within the bounds of the loose item in question. A hit or do on a loose item will cause some action to be carried out.

The popular loose items known to most users would probably be the ZZz, ESC, resize and move ones that appear in the caption bar's of may PE programs.

Application Sub Windows

There's not much to say about the application sub windows really. They are what's left of the primary or secondary window after borders, information sub windows and loose items etc have been removed. They are the areas of the screen that the program prints its output or allows input from the user and so on.

A graphics drawing program, for example, would use the application sub window to allow the user to draw whatever it is that they are drawing.

Pan and Scroll Bars

These are displayed if the data in an application sub window is too wide (pan) or too tall (scroll) to be displayed completely within the area of the screen set aside for the application sub window. GUI users on other system (Linux or Windows) will be familiar with the concept.

At first, these can be a nightmare as a 'DO' within the scroll bar (or pan bar) will split it and you then end up with two separately scrollable (and/or pannable) windows within the application window. Could be useful at times I suppose!

Sprites, Blobs and Patterns

A SPRITE is a picture that appears on the display somewhere. A pointer is just a sprite that is moved around the screen. Sprites may be drawn to look like text, for example, in logos and programmer's names etc, or they may be small pictures to represent some function of the program.

A BLOB is part of a sprite and holds only data that defines the shape. It has no colour information at all. The PATTERN is the part of the sprite that holds the colour data. Why separate them like this? I suspect it was to save memory - why bother having sprites defined with the same shape, just different colours - by defining the BLOB once and the PATTERNS for the colours, you save repeating the blob data - perhaps?

Blobs and patterns can be used independently of sprites though.

Border

The border around the primary and secondary windows, and indeed any other object, is optional and up to the programmer. However, most programs use borders.

When you move the pointer over a loose item, a border may appear around it to indicate that you can carry out some form of action if you were to hit or do the loose item in question. Once the pointer is outside the loose item boundary, the border may vanish.

Shadow

The shadow for a window is drawn down the right side and along the bottom. It is optional and entirely at the discretion of the developer. When in use, a shadow gives the impression that the window is hovering above the desktop. The shadow is outside the outline and does not register hit or do actions. It is purely decorative.

More Useful Utilities From George

The GWASL assembler that George Gwilt wrote has been used as the assembler of choice throughout this long running series. George has come up trumps again with another utility that allows the easy generation of assembler code that defines a WMAN windows definition (more on this later) and I've been testing it out. Unfortunately, my holiday got in the way and I have a new version of the utility and GWASL to test out at the moment.

I'm sure that these programs will soon be available from George's usual repository of fine code. In addition, I shall be trying these utilities out myself and reporting back.

WMAN Windows Definition

As mentioned above, WMAN is slightly more involved than the bare bones PE in as much as it carries out a huge amount of work on your behalf. This is all work that you would have to write into each and every program you write using the PE/WMAN system (hereafter known collectively as the PE or the Pointer Environment) but in order to take advantage of all this hard work, you have to set things up in a standard manner.

If you look back an issue or so, you will notice that up until now, all my PE test programs simply opened a console and set an outline before entering the main loop to read the pointer, act upon it, repeat as necessary. Obviously, my test programs were small and insignificant - but even though, they could benefit from a bit more added 'sparkle'. The WMAN routines make this possible.

The first thing we have to do is create a definition of our window in memory. This will be in a standard format and when done, we call a WMAN routine (WM_SETUP) to initialise the various internals required to make our window work under WMAN. Let's now take a look at the standard definition as required by WMAN.

Standard Windows Definition

So, now you know what all the bits in a window are, we can get right in and start discussing the standard way we have to define a windows and all its decorations. Let's take a look at one that someone else prepared earlier.

The following is extracted from a small utility written by Oliver Fink many years ago. The utility shows various bits of information about the running QDOSMSQ system. I have modified the original in a few places but the full credit must remain with Oliver. The code is in the public domain.

The start of the definition is the main window itself:

```
;Main window definition :
            dc.w  160              ; default window width
            dc.w  84               ; height
            dc.w  146              ; initial pointer x position
            dc.w  8                ; y position
```

So far so simple, nothing much here that we haven't met already. All we are doing here is telling WMAN how big our window is to be and where within the window the pointer is to be positioned when the window is first drawn.

The above positioning of the pointer is relative to the window outline. So in our window which is 160 pixels wide, the pointer is located 146 pixels along - nearly at the far right end. It is located 8 pixels down from the top. When drawn on screen, this places the pointer directly over the ESC loose item.

When the program is first executed, the PE attempts to position the main window on the screen so that the requested position of the pointer is superimposed on the current pointer position on screen. This prevents disconcerting jumps of the pointer every time you start up a new program.

You can see this in action if you move the mouse around on screen, note where it ends up, then EX a new program that uses the PE. You will see that the main window appears wrapped around where you last saw the pointer.

Next we define the attributes for our window.

```
            dc.b  $00              ; MSbit clear to call CLS
            dc.b  2                ; shadow depth
            dc.w  1                ; border width
            dc.w  0                      ; border colour (black)
            dc.w  7                      ; paper colour (white)
```

Again, there is nothing remarkably difficult here. Bit 7 of the first byte tells WMAN whether or not the window is to be cleared. Setting bit 7 says that the window must not be cleared. Following on, we define a shadow size for the bottom and right edges of our window.

Remembering back to our initial forays into the raw Pointer Environment, you may remember that we could have a different shadow depth on both of those sides, using WMAN, it appears that the shadow must be the same down each side. Oh well!

Note: The documentation says that for sub-windows the shadow depth should be zero. Best we stick to that advice. Remember, a sub-window is one 'embedded' within the main window. See application sub-windows or information sub-windows above.

Next, and finally for the main window, we have the definition of where the default pointer sprite for the window is to be found.

```
            dc.w  0                      ; use default pointer
```

This is one of my changes. In a need to reduce the amount of code in the magazine and also, to reduce your typing, I've modified Oliver's definition to use the default arrow pointer in the main window and in the application sub-window which will be defined below. Oliver had a custom sprite for the main window and another for the application sub-window. Both have been removed. The original file had this definition (don't type this in!) and a chunk of code to define the sprite to be used.

```
            ; DO NOT TYPE THIS IN!
            dc.w  sprt-*                 ; pointer to pointer sprite
```

You should be aware that all pointers in a window definition are word sized and relative to their own position in the definition block.

Now, that implies that all object lists must be within plus or minus 16KB of the pointer position, which might be a problem when there are a lot of objects and so on to define. To this end, if bit zero is set - an odd address - then that offset is used as a pointer to a long word which itself is a relative pointer to the object in question. Obviously, the word length odd pointer obviously has to be made even first, this is done simply by clearing bit zero.

In the above, if the Pointer Sprite above was defined a long long way away, we would see something like this:

```
            ...
            dc.w  sprt-*+1               ; ODD Pointer to a long pointer to
                                         ; our window's pointer sprite.
            ...
spr t       dc.l  Real_sprt-*            ; Long pointer to pointer sprite.
            ...
Real_sprt   ....                         ; Pointer sprite definition.
```

If any pointer is to something we don't need, then simply set it to zero. So, for example, instead of using Oliver's original '?' sprite (shaped like a question mark) we have defined this word pointer as zero and get the default arrow sprite instead. In this case, zero means 'use the default' but in other places, it means 'not used'.

Next to be defined are the attributes for all the loose items we will be using in the window. Starting with the easy bits:

```
; Menu item attributes
            dc.w  1                      ; Current item border width
            dc.w  0                      ; Border colour (black)
```

As before, it is simple. We define a black border 1 pixel in width. When the pointer is over any of our loose items, this border will be drawn around it to indicate that 'you can do something here'.

Following the border, we have the attributes for the loose items that are unavailable, available and selected. These attributes require 8 bytes each and define paper and ink (for text objects contained within them) and also blobs and patterns for the other object types. We are only using the paper and ink attributes, but the others must be there. We use zero to indicate 'not in use'.

NEXT QUANTA SPONSORED EVENT

The 25th Annual General Meeting will be held at the
Allesley Hotel, Coventry, England. CV5 9GP
On Saturday April 18th and Sunday April 19th 2009.
The AGM will be held at 2.00 pm on Sunday

All enquiries to Sarah Gilpin

secretary@quanta.org.uk

```
; Menu item unavailable
            dc.w  30                ; Paper colour (green/white stipple)
            dc.w  30                ; Ink colour
            dc.w  0                 ; Pointer to blob for pattern
            dc.w  0                 ; Pointer to pattern for blob

; Menu item available
            dc.w  7                 ; Paper colour (white)
            dc.w  0                 ; Ink colour (black)
            dc.w  0                 ; Pointer to blob for pattern
            dc.w  0                 ; Pointer to pattern for blob

; Menu item selected
            dc.w  4                 ; Paper colour (green)
            dc.w  0                 ; Ink colour (black)
            dc.w  0                 ; Pointer to blob for pattern
            dc.w  0                 ; Pointer to pattern for blob
```

In this example program, the loose items are never anything except available (and very briefly, selected) so the unavailable attributes are never used, but they still have to be defined. Oliver has chosen to use a paper and ink colour of 30 for unavailable loose items. That value gives a pleasant green/white stipple. Don't worry about it because you will never see it!

Following the loose item attributes, we have a relative pointer to the help window. In this program, we are not using one, so that pointer gets the value of zero.

```
            dc.w  0                 ; Pointer to help window
```

That is the end of the fixed part of the window definition. So far so good, there has been nothing too difficult yet. I wonder what is coming?

The rest of the definition block defines the repeating parts of the window definition. What exactly does that mean?

The documentation has this to say about the repeating parts:

To allow for a variety of different layouts within the window as the size of the window varies, part of the window definition may be repeated several times. The definition should be made in order of decreasing window size. The last definition which defines the smallest allowable window, should be followed by a word containing -1. If the top nibble of a layout size word is zero, then the layout may not be scaled. If it is %0100 then it may [be scaled].

So there you have it. The fixed part of the window defines the default layout for the window. That layout and all other possible ones allowed, need to be defined in the repeating part of the window definition.

A window can be scaled by WMAN if the definition allows for it. The scaling flag is the top nibble (4 bits) of the size words for the window layout. If the top nibble is %0000 then it cannot be scaled and if it is %0100 then it may be scaled.

Note: This actually shows up what I think is a contradiction in the documentation. There are other values that can be used in the top nibble, not only %0000 and %0100. More on scaling later in the series.

Scaling applies separately to the width and to the height of the different layouts. You don't have to scale vertically and horizontally, you can pick one, the other or both as desired.

For simplicity, and because I have not investigated scaling yet, Oliver and I will be sticking to non-scaled windows for now. Scaling will be a subject for a future article.

The following is the repeating parts for our single, non-scaling layout in our small program.

```
; Base of repeated part of window definition
            dc.w  160               ; Width for this layout
            dc.w  84                ; Height for this layout
; Pointers to definition lists
            dc.w  il-*              ; Information sub-windows
            dc.w  ll-*              ; Loose menu items
            dc.w  al-*              ; Application sub-windows
```

The above would be repeated for each and every different allowable layout for the window. Following on from the very last layout, the smallest allowed, we have the terminating word.

```
            dc.w  -1                     ; End flag
```

So we allow one and only one layout, which just happens to be exactly the same size as the default one defined in the fixed part of the definition block. It has three pointers at the end for the information sub-windows, the loose items and any application sub-windows that are required in this layout. Each layout will have it's own list and they need not be the same for each different layout.

We shall pause for breath at this point and discuss these lists in the next article. Hopefully, the above was not too taxing and I've explained it better that I ever had it explained to me!

Next time, we continue our look at the window definition by looking into the lists of objects attached to our window.

EasyPEasy

by George Gwilt

It seems that most people find writing programs for the Pointer Environment (PE) fairly, if not extremely, difficult. Probably the two main difficulties are understanding and producing a Window Definition, which lies at the base of all PE programs, and understanding how the PE software works.

There are various programs designed to help programmers. For those writing in S*BASIC there are EasyPTR and TurboPTR. I exclude the SuperBASIC version of QPTR because that does not rely on a Window Definition. For C programmers there is CPTR. For assembler programmers there is the set of macros in QPTR, written for the assembler Qmac, which help in producing a Window Definition suitable for assembly as relocatable code needing linking with other modules to produce the final executable program

I will now describe another way of helping assembly programmers. This is called EasyPEasy short for Easy PE Assembly. This is available on my website at

web.ukonline.co.uk/george.gwilt

The help given by this goes beyond that in the QPTR macros. A further difference is that the resulting source code can be assembled by either GWASL or GWASS without the need for linking.

The files included in EasyPEasy comprise Keys, Sprites, Subroutines, Examples and an explanation, or guide, in Readme.

An extract from the guide is given here:

* Start of Extract *

This guide to writing PE programs using Assembly Language contains the following sections:

A - Short Description of PE
B - PE Windows
C - Program structure
D - Files provided

A - Short Description of PE

The Pointer Environment (PE) consists of two parts, PTR_GEN and WMAN. The first of these is a CON/SCR driver which contains many extra Trap #3 routines.

The second is a set of vectored routines based on the new Trap #3 routines enabling a programmer to operate windows in a PE program.

A PE program is operated by means of a pointer, usually by means of a mouse.

B - PE Windows

At the basis of all PE programs is the Window Definition. This contains details of all the possible windows which can be used inside a program.

The definition starts with a fixed section detailing amongst other things the maximum window size, its border size and colour and its paper colour. This section also has a pointer to the sprite to be used as its pointer.

The fixed section is immediately followed by a set of at least one repeated sections. Each of these has a size which must be less in at least one dimension than that of the previous section and must not be larger than the maximum size in the fixed section. Each repeated section has pointers (which may be null) to lists of loose items, information windows and application windows.

The loose item list consists of a set of sections which detail the size, position and type of item (for example text or sprite). Each item also contains a pointer to its action routine.

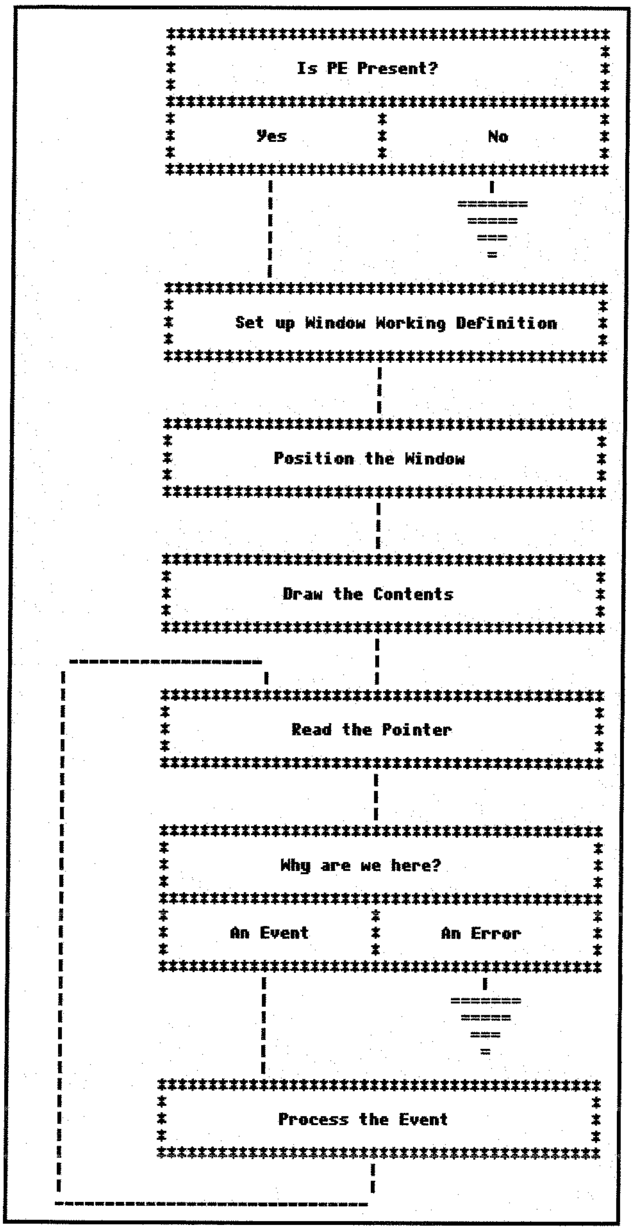
Similar information is given in the lists of information windows and application windows, though for the former there are no action routines.

Associated with the window definition is the

64-byte status area which is immediately followed by a set of bytes, one for each loose item, which constitutes the loose item status block.

C - Program Structure

The following flow chart shows the structure of a simple PE program.



D - Files Provided

To help in the production of PE programs a number of files are provided. They are:

A. KEYS

keys_pe keys for PE Trap #3 routines keys_wdef keys for the window definition keys_wman values of WMAN vectored routines keys_wstatus keys for the status area keys_wwork keys for the working definition qdos_pt pointer interface keys

B. SPRITES

The sprites given here are system sprites for mode 4 as well as a few extra sprites, such as the hand sprite, for all modes.
csprc_bin sprites which can be LIBd csprc_sym_lst the names of sprites in csprc_bin

C. SUBROUTINES

peas_bin subroutines to be LIBd peas_sym_lst names of subroutines in peas_bin
The subroutines are defined here:

GETSP returns in A0 the address of an area of D1.L bytes. If there is an error, the program commits suicide.
No other registers are used.

MOVE causes a move, then clears D4 and D0. No other registers are used.

RECHP returns to the heap the area with base address in A4. No registers are used.

SET_AP sets an application window menu
At entry:
D1.W = Number of items
D2.W = Number of bytes between items
A0 -› Items
A1 -› application window
A4 -› working definition
No registers are used

SLEEP sets the program to a button which contains the name of the program and is placed in the button frame if there is one or at the top left of the screen if there isn't. A HIT (left click or SPACE) will cause the program to waken. A DO (right click or ENTER) will cause the program to waken if there is a button frame or causes a move if not. This enables a user to set buttons at different places on the screen if there is no button frame.
At entry:
D1.L = size needed for the button (ww0_1)
D2.L = size for main window (ww0_0)
A2 = WM vector A4 -› working definition
On exit:
D1-3 are used A0 = channel ID
A1 is used
A2 = WM vector
A3 = Window definition
A4 -› working definition (may have changed)

SUI quits

D EXAMPLES
Seven examples are given, Ex0 to Ex7 excluding Ex5. For each example there are three files, the source code (_asm), the window definition (w_asm) and the assembled executable program (_bin). Thus the files for Ex0 are Ex0_asm, Ex0w_asm and Ex0_bin. Each program shows some aspect of PE programming.

Ex0 shows four common operations move, resize, sleep and quit.

Ex1 shows how to set up the menu of an application window from the directory of flp1. It also shows how to cause a sleep by CTRL/F1 without a corresponding loose item.

Ex2 shows a how to set a secondary window. Items in that window can be selected. A right click in the area containing the items (but not on an item) causes the first window to reappear. Clicking Print will now print the items selected. This window can be set to a button by pressing CTRL/F1.

Ex3 shows drawing on the screen. Clicking OFF or pressing O allows drawing. The colour can be changed by pressing W, G, R or B. The display is cleared by pressing C (even when Clear is "not available"). Drawing ceases on a right click, pressing ENTER or O. This program does not go to sleep on pressing CTRL/F1.

Ex4 shows how to drag an object inside an area. Holding down the left mouse key causes the pointer to disappear and allows dragging until the mouse key is raised.

Ex6 shows how small explanatory windows can appear if the pointer is left on a loose item for a sufficient period of time. This program can be buttonised by CTRL/F1.

Ex7 shows the resizing of a secondary window. This is not exactly the same as resizing the primary window.
Note that all examples can be assembled by GWASL as well as by GWASS. QPTR

* End of extract *

Window Definition

None of the files in EasyPEasy help to produce the Window Definition. How then should a programmer produce this? One way is by "direct generation" as the manual for QPTR calls it. That is by setting the numerical values to a set of DC.W instructions. However, when it comes to producing the Window Working Definition, it is necessary to know the amount of space required. This can, laboriously, be calculated from the Window Definition but it is advantageous to have some better method of finding this. The macros of QPTR do it. Another way is to use SETW.
SETW is a program which produces Window Definitions for assembler use.
It may appear easier to prepare a window definition without the use of a program whose operation may not be immediately obvious and so has to be learnt. But SETW has certain advantages which should not be overlooked.
1. The sizes of working definitions are provided.
2. The definitions of text items are provided.
3. All system sprites and a few others are provided.
4. The windows produced are viable.

The last advantage will certainly be appreciated by anyone who has attempted to produce a window definition without the help of a program such as SETW. The chance of a window being accepted first time by the PE software is remote. When something goes wrong the message "out of range" is peculiarly unhelpful since it may refer to any of the large number of constituents of the window.

Still QLing, despite the odds

by Wolfgang Lenerz

I'm one of those who still use their QL (actually QPC) every day or so. Indeed, I've written a „business suite" which allows me to keep track of my cases, and which also handles the (very modest, see below) accounting needs of our office. I don't pretend to have written something revolutionary or particularly well done. It's just that the software is tailored exactly to our needs and to my way of working. It's thus understandable that I'm pretty loathed to give it up and use some PC software that an outside firm is trying to push onto our office. Since I determine what software our office uses for anything common to all of us, that switch hasn't happened, at least until recently. However, cost factors have caused me to rethink my strategy.

To explain how this came about, a small word of how out accounting is handled here. Like anyone self-employed, we have to have accounting books, double entries, ledgers and tutti quanti. This actually can take up quite some time that I could usefully spend on something more worthwhile (like doing some QLing). Moreover, accounting regulations change over time and I don't really fancy having to keep up with something like that every day. So we use the services of an accounting firm (which is attached to our bar and thus has competitive prices). We supply it with raw data, consisting of a day-to-day record of what money was paid/received where and for what, and that firm then generates all the correct books in the correct format. It also automatically handles depreciation and write-off of our equipment, formally audits the books and thus our accounting and also certifies it. The certification entitles us to a 10% reduction of taxable income. So you could say that apart from our recording the every day flux of incoming and outgoing funds, our own accounting needs are pretty slim, since more or less everything else is handled by the accounting firm. My own accounting program is of course tailored to that and generates a (monthly) report for the accounting firm with the required information. It also handles some other queries and data (e.g. what is the financial status of a case, which partner generated what kind of income etc...) which aren't strictly necessary for accounting purposes only, but are necessary (or at least useful) for our office.

Anyway, for the formal accounting, I thus generate a report that is sent to the accounting firm. The accounting firm takes that information and turns it into the correct books etc. Up until last year, the way I sent this information was ...(ahem)... on paper. Yes, strange as it may seem, I typed that info into my computer, and then printed it and sent the printed sheet to the accountants. They then retyped this information (!) and did what they had to do with it. Several years ago I thought that this was a pretty stupid way of doing things, after all I already had this information electronically why could I not send this to them electronically? So I inquired after that and was informed that this was, of course, possible – but to do that I would have to use their own accounting program running on a PC.

Of course, that was a no-no for me, though I was told that, for retyping the data, they actually had it retyped in India (!), so they probably scanned the pages I sent them and emailed them to India. Pretty inefficient if you ask me... Anyway, not to be deterred I inquired further and it turned out that I didn't really have to use their software, "all" that was required was that I send them the information in exactly the same format as that used by their own software. OK, so let me have the specifications for that, I asked – after all there must be many who, like me, use another program than the one sold by the accountants. This actually was confirmed to me (apparently, many used a simple Excel datasheet for the day-to-day records). I was promised that I could have the specifications, and ... I really got them. I reckoned that with these specifications I could generate, from within the QL, the information in the correct format. I was wrong. Have you ever read specifications that were drawn up specifically so that they are NOT understandable? Well, here I had a perfect example of something like that. It was quite clear that these specifications were set out in such a way as to require so much study just to understand them as to make the effort simply not worth the while. So I just abandoned that idea and kept sending my information on paper.

Of course, this couldn't continue forever, even the accounting firm noticed that this was an inefficient and (for them) costly way of doing things. So, to get to the die-hards like me, what they did was to increase the (yearly) prices for the "paper" version of their services whilst decreasing those for the (yearly) software version thereof. For a few years I just gritted my teeth and paid the increased prices only because I wanted to be able to continue using my software which, as I said earlier, is exactly tailored to our needs and way of working. However, after a few years, the price differential between the "paper" version and the software version became just too important – we're talking about several hundred euros a year here, and even if you're used to a certain way of doing things, if it becomes that uneconomical, you just have to amend your ways.

So, last year I decided to bite the bullet and get their software. After all, I presumed it would be OK and have more or less all of the functions I'd need. Oh boy, was I wrong.

As soon as the very nice lady from the accountants came to train me on the use of the software (all of a few hours worth) I smelled disaster: I was faced with a totally rigid interface having only the most basic of functions and obviously bolted on to something right out of DOS times. I know that we lawyers are often a pretty backward and conservative bunch, but here I had met my masters. Shudder. An example: in the (modern) PC world, when you want to delete something in a field or a control, you just mark (highlight) it with the mouse and then hit delete.

You can't here. You have to delete every single character. Re-shudder.

There was just no way I was going to use that. So what to do? I thought that there had to be some way to get the information from my QL into the accounting software, but how? At the same time as this happened, I'd also started to have a look at "Java", a modern programming language of the object oriented kind, and I thought that I might be able to knock something up with that. But what?

As an aside: Opinions about Java vary enormously, it has the (often undeserved) reputation of being very slow. I don't really want to go into that debate here. I find it an elegant language with one advantage: it is pretty safe and leads you to write programs that have fewer bugs from the onset. To my mind, that is a great advantage. However, Java is also a language that isn't really close to the machine – there is, for example no way to set the screen resolution from within a Java program.

Anyway, I really didn't want to have to re-dig through the specifications for the accounting program to see whether I could generate from the QL a file as if it was generated by the accounting program, so I had to find a way of getting my information directly from the QL side into the accounting program. I lucked out there: Java has a "robot" class. (For those who don't know object oriented programming, a class is a self-contained set of program routines and data that are supposed to the entirely reusable). The "robot" class enables you to simulate mouse clicks and keyboard hits. You can thus tell it to put the mouse cursor at a certain point of the screen. You can also tell it to simulate a right or left mouse click (mouse button pressed and mouse button released). You can also direct it to simulate a keyboard hit – it will thus insert a character into whatever field or control has the keyboard focus.

OK knowing that, the whole exercise became pretty simple in theory: the program would simulate me. It would read the data from a file that is generated from the QL side of things and input it into the accounting program's form as if I'd typed it in. I thus amended my QL accounting program so that it printed the information no longer to a printer but to a file, which is always called "c:\main_int_storage.txt". (No, don't ask me how I came up with that name, I have no recollection of it). So I always send the QL accounting report to that file.

This information is in a very simple structure : it is subdivided into "records", each record being just one transaction (money paid or received). Each record starts with the words "**RECORD START**" and ends with the words "**RECORD END**". Each record contains the same kind of "fields" : first the "PCG" number. The PCG is the general accounting plan – each transaction is posted to a determined account identified by that number (PCG code). Then the record contains the date of the transaction, the amount with and without VAT, the title of the transaction (e.g. "Paper" if I bought reams of paper) etc. Each field in turn is introduced by a label, such as "**DATE**" for the date, or "**SORTIE**" if the money went out of our pockets and "**ENTREE**" if it came into them (I should probably have done it differently and generated XML tags). All of the data is in simple ASCII text. Thus, this is a pretty simple text file of a clearly defined format. This text file is then read in by the Java program which does some basic checking (more of this later) and inputs the data into the accounting program.

I don't know whether the editor will include the listing of the Java program here – I don't pretend that it is an example of good programming. Indeed, when I went back to it to have a look at it for the purpose of preparing this article, I wasn't surprised to read that I had commented it as "this is a pretty awful hack". I wrote it rather quickly, pretty inefficiently and not very elegantly – I never really went back to improve it, once something worked. However, it seems pretty bug free and does what I want - and I can't really be bothered to change something that works. It is called, fittingly but unimaginatively, "Inputter".

Developing "Inputter" wasn't too difficult. I used an IDE (integrated Development Environment) called NetBeans IDE. This provides you with a development area where you can draw the window of your application-to-be, put the buttons into it and then write the code that is supposed to happen when the user pushes the buttons. It also allows you to run the program, set breakpoints in it, inspect variables etc. (something like that is REALLY missing in the QL World). Again, I don't want to discuss the merits or not of NetBeans as compared to other IDEs such as Eclipse etc. It just happens to be what I used and you will find some evidence of that in the code of the Java program as the environment generates some code itself. (As an aside, I even know people who always write their Java code within a simple text editor - one of these is one of the best professional programmers I know!).

When the Inputter program is started, it shows just three buttons: "Start Aidavocat" (executes the accounting program), "Load data" and "Insert data". These three buttons are supposed to be pushed one after the other and actually show how the program itself was developed.

Indeed, first of all, I tried to find out whether from within a Java program I could actually start up another program, or whether I would have to start the accounting program first and then my program. Java provides a "runtime" class however, which allows you to execute another program. The line

```
"p=r.exec(comptaDir+"AIDAVOCAT.exe"+
comptaDir+"Aidavocat.lbr")"
```

calls up the "Aidvacoat.exe" accounting program and passes it "Aidavocat.lbr" (its accounting database) as a parameter.

So, once this is done, the accounting program is set up and running. Then I push the next button of the "Inputter" program - this loads the QL generated data into the Inputter program. Nothing fancy here, on the contrary, it is all pretty unspectacular and dull and not very efficient. The procedure "file_io_open" first of all tries to open and read a file called "comptes_existants.txt" (= existing accounts). This is a simple text file containing all of the PCG codes the program should know about.

This deserves some explanation: As mentioned earlier, each account in the general accounting plan (PCG) is identified by a number code. Most of these are pretty standard and all accounting programs must know about them, and do. However, you are also allowed to create your own sub-account numbers, within this framework. For example, account number 760000 generally identifies fees received with a normal VAT rate. Within that account, I can create more sub-accounts, such as 760001 - fees received by me, 760002 - fees received by another partner etc... Of course, my QL accounting program "knows" all about these sub-accounts, but the other accounting program didn't know about them at first. So I had to create these sub-accounts first in the PC accounting program. This is pretty normal and has nothing to do with the awkward handling of the accounting firm's program, you have to teach every accounting program the quirks of your own firm.

If you attempt to introduce a transaction with an (as yet) unknown account number, such as 760001, then the accounting program complains, as it should. No accounting program can create those sub-accounts all by itself, it needs your input for that creation. SO when you try to use an unknown account, the program has to complain. This complaint is materialized, in this case, by a window that pops up and tells you that the account you are trying to use is unknown. Again, this is as it should be. However, in an automated system such as mine, where the computer fills in all of the fields, such a new window will play havoc with the inputting, since the Inputter doesn't know about this new window and cheerfully tries to fill in fields that are no longer accessible. To avoid that, I keep a simple text file with a list of account numbers the accounting program already knows about. This list is read by the Inputter program. Then, for each record in the QL file read by Inputter, it checks the PCG number of the transaction against the known PCG numbers. If the PCG code of the transaction is NOT known (and thus as yet unknown to the accounting program), then Inputter doesn't even try to input this record. It sets it aside, and once it has finished, it writes all of the unknown records out into a log file called "c:\main_int_storage2.txt". More of this later.

If the PCG is known and the record seems OK, Inputter keeps the respective information inside a very simple array. At the end of the reading process, it has one array with all records that are to be input into the accounting program and one array with the unknown records.

The third button actually causes Inputter to do the inputting work. First of all, it brings the accounting program to the front of the screen. The accounting program always takes up the entire screen when it starts up. So I put the mouse at screen co-ordinates (0,0) and "click" - this causes the accounting program to come to the front, just like a program lying behind another one pops to the front when you click on it. From now on, every mouse click and keyboard tap will happen inside the accounting program. After that, Inputter clicks on some buttons inside the accounting program, to bring up the input form to be used. Again, this is done by putting the mouse cursor at the screen co-ordinates where the respective buttons for this can be found in the accounting program. Since the accounting program always takes up the entire screen and since I run this at a determined screen resolution, the location of these buttons never changes. I used a freeware program (I can't remember which one) to find out the exact positions of the buttons. This position is simply programmed into Inputter itself.

Now I put the mouse cursor simply smack in the middle of each button and "click" on it. Once the input form is up, the program takes each record in turn, and inputs the information into the form. This is done by clicking into each field, cumber-

somely deleting everything in there (by repeatedly pressing and releasing the "delete" key) and inserting the new data. At the end of the record, Inputter "clicks" on the "validate" button to accept this entry and starts over with the next record.
Once it's finished inputting all of the data, Inputter brings itself to the front of the screen and tells me how many records it input. This allows me to see at a glance whether all of the records saved by my QL program were input into the accounting program (the QL program tells me how many records it wrote out). If necessary, I check with the "c:\main_int_storage2.txt" log file to see what went wrong. Mostly this is simply that I used a sub-account the PC accounting program didn't know about. - it is astonishing how many sub-accounts I have created over the years in the QL program. Once I have identified the accounts the PC accounting program doesn't know about, I create them in it, rename "c:\main_int_storage2.txt" to "c:\main_int_storage.txt" (i.e. the normal input file for Inputter) and rerun Inputter so that it finishes inputting the information from that file into the accounting program, too. For that, the log file has exactly the same format as the normal input file... Needless to say, nowadays, where I have been using this system for over a year, the number of unknown PCG codes has decreased dramatically, so most of the time, the program runs without a hitch.
I generally let Inputter run when I'm doing something else somewhere else. Indeed, whilst Inputter is running, you can't use the computer for anything else since the program is steering the mouse and inputting characters. It clicks at a certain place on the screen without knowing what is underneath. So if you open another program over the one where the input is supposed to go, the input will go into that newly opened program, which definitely isn't what you want. Hence also a word of warning: if you ever intend to use something similar, make a large number of test runs with very few (but different) data during each run: it is nearly impossible to stop the Java program whilst it is inputting data, since the mouse moves all over the screen and characters magically appear. Even the three fingered salute (Ctrl-Alt-Del) which brings up the task manager under Windows is dangerous, since you could inadvertently close down all kinds of programs with it as Inputter will cheerily click in it if it happens to have its window at a place Inputter is supposed to click....
Once the process is finished, I have input my data into the PC accounting program and can send it to the accounting firm. All in all, I'm pretty happy with the solution I found – I pay less money but still use my old QL program, whilst giving the accountants what they want.
PS. as a final word, if you look at the code you will see some references to Linux. I actually wrote "Inputter" under Windows. When I ran it under Linux (the PC accounting application then ran under "Wine"), I notice that a mouse click on a certain screen position didn't click at the same place in the form.... Grrr.... I'll have to investigate and probably just change the entire button emplacement data. QL forever!
[Editors note: we have started laying out the program. It was soon clear, that it would fill many pages, even in very small print (8pt or smaller) so we thought, we'd rather put it onto the qltoday website and on the next cover disk. If you want us to print it, please let us know...]

QUILL VIEW - last minute News

by Geoff Wicks

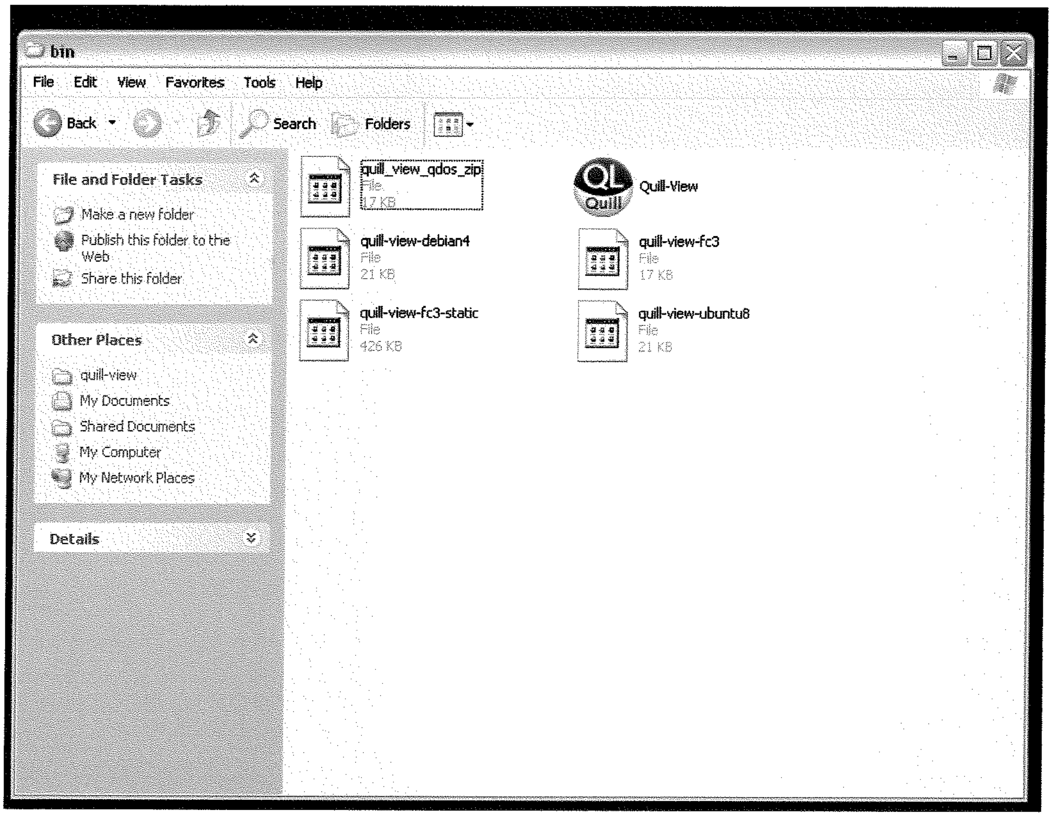
Mikael Ström has written a viewer to display Quill_doc files in non QL operating systems. In his own words:

"Quill-View was written because I got tired of firing up QL2K every time I needed to read a Quill document. Writing the code was more complicated than anticipated, and there are still a few things that need to be done. However it now properly formats all documents that I use."

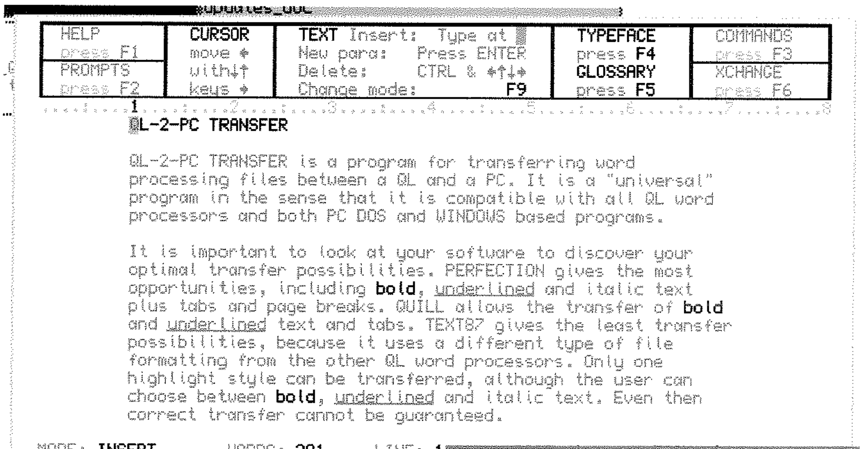
QL Today received its copy of Quill View too late to be included in the main news section (the editor's fault and not Mikael's), and we only have time and space for a short review. We would welcome a more thorough review from any reader who is interested.

Quill-View currently runs on Windows XP and Vista, QL/QDOS, Debian 4,Ubuntu 8 and Fedora 3 through 9 (and Red Hat).

QL Today received its copy of the program as a zip file containing versions for several different operating systems and documentation in Quill_doc, HTML and text formats. It is not necessary to install the program on a PC. I expanded the zip file to a stick memory and the Quill-View.exe application was ready to use.



To test the program I used material I had written to test QL-2-PC Transfer. All I had to do was to transfer a Quill_doc file to a PC medium, and then drag and drop it to the Quill-View icon and click. The text is then displayed in the default browser. Below you can see the original Quill document and the browser text.

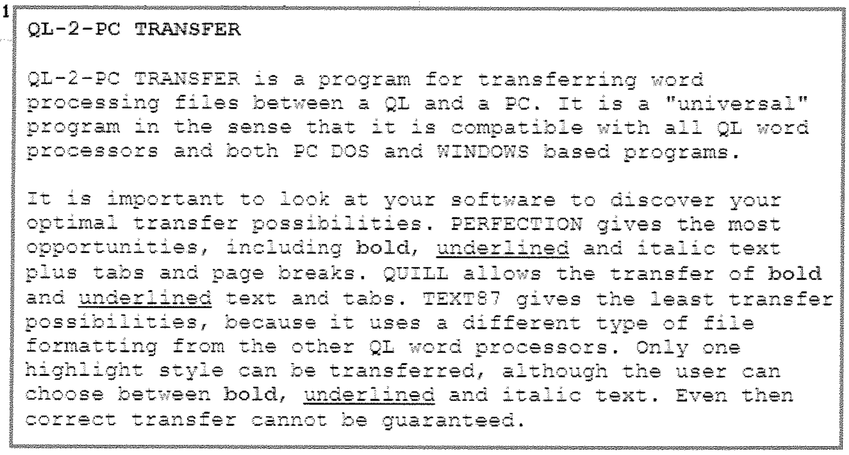


```
HELP       CURSOR    TEXT Insert:  Type at      TYPEFACE   COMMANDS
press F1   move      New para:   Press ENTER    press F4   press F3
PROMPTS    with↓↑    Delete:     CTRL & ←↑↓→    GLOSSARY   XCHANGE
press F2   keys      Change mode:          F9   press F5   press F6
.......1.........2.........3.........4.........5.........6.........7.........8
    QL-2-PC TRANSFER

    QL-2-PC TRANSFER is a program for transferring word
    processing files between a QL and a PC. It is a "universal"
    program in the sense that it is compatible with all QL word
    processors and both PC DOS and WINDOWS based programs.

    It is important to look at your software to discover your
    optimal transfer possibilities. PERFECTION gives the most
    opportunities, including bold, underlined and italic text
    plus tabs and page breaks. QUILL allows the transfer of bold
    and underlined text and tabs. TEXT87 gives the least transfer
    possibilities, because it uses a different type of file
    formatting from the other QL word processors. Only one
    highlight style can be transferred, although the user can
    choose between bold, underlined and italic text. Even then
    correct transfer cannot be guaranteed.

MODE: INSERT        WORDS: 281      LINE: 1
TYPEFACE: Bold
```



```
QL-2-PC TRANSFER

QL-2-PC TRANSFER is a program for transferring word
processing files between a QL and a PC. It is a "universal"
program in the sense that it is compatible with all QL word
processors and both PC DOS and WINDOWS based programs.

It is important to look at your software to discover your
optimal transfer possibilities. PERFECTION gives the most
opportunities, including bold, underlined and italic text
plus tabs and page breaks. QUILL allows the transfer of bold
and underlined text and tabs. TEXT87 gives the least transfer
possibilities, because it uses a different type of file
formatting from the other QL word processors. Only one
highlight style can be transferred, although the user can
choose between bold, underlined and italic text. Even then
correct transfer cannot be guaranteed.
```

To the left, the document as displayed in Quill, below the result of Quill-View.

If you press Alt at the same time as you drag and drop the document is automatically loaded into notepad or whatever program you use for displaying text files. (Notepad is not very useful as it does not support all of Quill's formatting.)

I also tested a Quill file containing accented characters and these were correctly transferred. There are one or two Quill features that are not supported such as soft hyphens, but these are often not supported in other transfer programs.

At the moment Quill View is still being developed, but the latest (beta?) version can be downloaded from a special Quill View page on Jimmy Montesinos' jadiam site.

http://www.jadiam.org/ QL2K/ QuillView

Jimmy also informs us that he has now released QL2K Build 101. This version enhances Vista support and is available in 32 and 64 bit editions. There is also improved sound support. There is also a new version of QLAYT.

Jadiam is Jimmy's company and this has just opened a new corporate website. At the moment the non-QL areas of the site are only in French.

RWAP Services NEWS!

**** We have moved ****

See our updated address details below.

We have also acquired more brand new Sinclair QL membranes and another stock of Epson Stylus Colour 850 inkjet printers, so if you need a better printer for your QL, give us a shout.

More news is always available on our website: www.rwapsoftware.co.uk

We are also looking to produce some new hard disk interfaces for the ZX Spectrum and have a few little projects on the drawing board.

Our websites:
http://www.rwapservices.co.uk (General site)
http://www.rwapsoftware.co.uk (Sinclair computer second hand and new items)
http://www.rwapadventures.com (Adventure Programs)
http://www.internetbusinessangels.com (Guidance on setting up online businesses).

New Products!

QWORD 1.0 NOW WITH DIGITAL SOUND ON QPC2!

The wait is now over! Q-Word version 1 is finally available!

Platforms:
QPC/QXL, Q40/Q60, Aurora (with SGC)

Prices:

All versions without P-Word	£20.00
All versions with P-Word	£30.00

Notes:
Q-Word DOES NOT require SMSQ/E with GD2 support -OR- SMSQ/E at all on the Aurora or Qx0 machines. It works on the highest colour depth everywhere regardless of Operating System.
The Aurora version is available on either HD or ED disk. For the latter add £1.00 to the price. ED version is uncompressed and can be run directly from the floppy. All other Floppy versions are compressed. QPC/QXL version comes on CD. Non CD versions DO NOW support digital sound on QPC2

TALENT COMPUTER SYSTEMS for Windows

For QLers that run Windows or with incompatible hardware for Talent Games, we now have re-released these adventures so that they can run on your Windows-equipped PC. No Emulator, floppies, microdrive backups etc. required, just a one-click install! Of course the full QL line is still available! (See side column)

Talent Games for Windows ea. £ 10.00
(Each Game includes a runtime installation of QLAY-2 by Jimmy Montesinos)

Games Currently Available from www.rwapadventures.com

The Lost Kingdom of Zkul
West
The Prawn
Return to Eden

Replacement Sinclair QL Keyboard Membranes

We always have a stock of brand new Keyboard Membranes (and keyboard parts) for the original Sinclair QL, so if you have some keys which no longer work, just give us a call.

Cost is only £18.50 plus £2.75 post and packing.

Second Hand Items - Huge Range Available

We stock a wide range of books, hardware and software for the Sinclair QL, Z88 and ZX Spectrum, including disk interfaces, memory expansion and microdrive cartridges. If there is anything you need - have a look at www.rwapsoftware.co.uk (or ring us with details of your requirements).

We are always happy to help.

Old Favourites!

Utilities

SBASIC / SuperBASIC Reference Manual on CD	£ 20.00
Sidewriter v1.08	£ 10.00
Landscape Printing (EPSON printers)	
ImageD v1.03	£ 10.00
3D object generator	
Q-Help v1.06	£ 10.00
Superbasic On-Screen help system	
Q-Index v1.05	£ 5.00
Keyword-to-topic finder	
ProForma ESC/P2 Drivers v1.04 for ProWeSs	£ 8.00
Printer Driver	

Applications

Flashback SE v2.03 (upgrade only)	£ 5.00
Database	
QL Cash Trader v3.7	£ 5.00
Accounting/Finance	
QL Payroll v3.5	£ 5.00
Accounting/Finance	
QL Genealogist v3.26	£ 20.00
Genealogy	
Genealogy for Windows	£ 50.00
QL Genealogist to Windows version upgrade	£ 25.00
QL Cosmos v2.04	£ 5.00
Planetarium	
Q-Route v2.00	£ 25.00
Route Finding	
Upgrade from v1.xx	£ 5.00
Britain map v1.11	£ 2.00
BIG Britain map (needs 2Mb) v2.03	£ 5.00
Various Britain Area maps (ask for details)	ea. £ 2.00
Ireland map v1.00	£ 5.00
Belgium map v1.01	£ 2.00
Catalonia map v1.03	£ 2.00
P-Word UK English Dictionary (500.000 words!)	£ 15.00
Dictionary	

Leisure

Return to Eden v3.08	£ 10.00
Adventure	
Nemesis MkII v2.03	£ 8.00
Adventure	
The Prawn v2.01	£ 8.00
Adventure	
Horroday v3.1	£ 8.00
Adventure	
West v2.00	£ 5.00
Adventure	
The Lost Kingdom of Zkul v2.01	£ 5.00
Adventure	
All 6 games above	£ 25.00
D-Day MkII v3.04	£ 10.00
Strategy/War Simulation	
Grey Wolf v1.08	£ 8.00
Graphical Submarine Simulation	
War in the East MkII v1.24 (upgrade only)	£ 5.00
Strategy/War Simulation	
Open Golf v5.20	£ 8.00
Sports Simulation	
QuizMaster II v2.07	£ 5.00
Quiz	
Stone Raider II v2.00	£ 5.00
Arcade Game	
Hoverzone v1.2	£ 5.00
Arcade Game	
Deathstrike v1.5	£ 5.00
Arcade Game	
Flightdeck v1.0	£ 10.00
Flight Simulation	
All 6 games above (Open Golf, QuizMaster II,Stone Raider II, Hoverzone, Deathstrike and Flightdeck)	£ 28.00

Notes on Software requirements
The following programs have a minimum SGC card requirement: P-Word, Qword, Big Britain MAP for Q-Route

RWAP Services
3 Dale View Court, Fulford, Stoke-On-Trent, Staffordshire ST11 9BA TEL: (+44) 1782 398143 From the UK Dial: 01782 398143
Website: http://www.rwapsoftware.co.uk
Email: sales@rwapsoftware.co.uk

We Accept Payment using:
PayPal VERIFIED NOCHEX EMAIL MONEY PREFERRED PAYMENT (Cheques in £ sterling made payable to R. Mellor)

QMENU Upgrades & Updates

by Jochen Merz

As announced in the previous issue of QL Today, QMENU Version 8 exists and has been shipped to everybody who ordered it.
Actually, it is V8.01 now and a new section QMENU Version 8 has been added to the JMS Update site for all the users who upgraded to V8 (I hope I have not missed any).
Of course, customers without a login ID have no access to their updates, so if you are NOT registered, please send me the login ID you would prefer and your password, and I will set up an account for you.
This service is free of charge, so why not use it? Which reminds me to say "Thank you" to Bruce for hosting this service!

For users who have not upgraded I have left the update download possibility to download the last version 7 (it was 7.68, created at Eindhoven in Summer 2008).

If you wish to place an Upgrade order to Version 8, you can do it through the SMSQ homepage of J-M-S:
http://SMSQ.J-M-S.COM

You will find links which lead you to a secure order form ... and the discounted price for QL Today readers (not just for upgrades but also for new orders) remains until March 2009.

A new feature of QMENU V8.01 is the possibility to specify the number of lines in the VIEW_FILE menu. You could always specify the number of characters, so why not... looks good and seems to be working well... one of the items on the wish-list from Bob Spelten.
Another bug, reported by George Gwilt, in SYSTEM_SELECT (just visual, nothing crashes!), has been fixed too.

I am thinking about a new QMENU menu which was suggest by François van Emelen: The idea is to show values in two columns next to each other, the right column being editable. For example, to show results, or to be able to edit fields of a database (e.g. left column field name, right column editable value).
I think, the best way would be to have two string arrays (one for the left column, or field names and one for the right column).
Optional flags could specify if the right (or maybe left or even both columns) are editable ... or would it make sense to add a status array to set the editable flag individually for each item?
I am in the stage of planning, so ideas and feedback are most welcome!

Please send your emails to **SMSQ@J-M-S.com** or use the contact form on the J-M-S homepage.

The Final Byts of Wood?

by Roy Wood

So, there I was, last issue, saying I was not going away and then, suddenly, everything changed. I suppose, in some way, I knew it had to but you do tend to keep putting things off and that really was the key to the problem I had. When, as a trader, you get one or two orders in a three month period you think, 'well I will do that tomorrow' and that turns into after the weekend and then next week, you know that is not the right way to do things. I had two orders for QPC2 sitting on my desk for ages and one of them I had not even taken the money for. The customers were patient, as most QL-people are but, in the end, I had to give myself a big shake and get on with it. It was at this point I decided that it was not fair on the customers - few that they are - and decided QBranch had to shut its doors.
I was reminded of a scene at a German QL show back in the mid 90's when Albin Hessler came up to Jochen and I and, almost tearfully, said he was going to have to stop programming for the QL. All things come to an end (except the M25 or the Paris Peripherique).
The problems with the last issue of QL Today were of the same root. Because it was the first one of a new volume I had to go through the subscriber list and take out the ones which had not re-subscribed. I had to send out reminders to some people who, I was sure had just slipped up. After that I needed to get a final total so I could

tell Jochen how many I was going to need. As most of you know I run the local branch of the British Sub Aqua Club as Diving Officer and have a very active rock n roll band. I compounded this by getting involved with a campaign to stop the local council turning Brighton Marina into a housing estate (still ongoing that one) and I have my family as well so I ran out of hours in the day. In the end the people who clamoured loudest for attention got it and I forgot to deal with the QL stuff. I apologise.

So I took a deep breath and decided this would be the last issue I would advertise in and after Christmas I would completely stop all QL business. Not that there was that much anyway.

It has been a good 12 years running the company and I have had a lot of fun with a lot of good people. I hope to stay in touch with a lot of them and I will still be using QPC2 on my PC. At the moment I am not sure if this will be the final column. As I said last time it has become increasingly hard to write copy about purely QL subjects and I have not been able to fill a whole column with it for a while now.

What Happened To All that Promise?

Well, some of you may be wondering what actually happened with the QL. I suspect it will not be many of you though. It should be fairly obvious that the progress of development stalled some years ago and, certainly for the last couple of years, no new products, updates or program versions have been issued. Lack of new stuff has lead to a lack of things to give talks about at shows and this, in turn, led to low attendance at those shows and their final demise. Is there anyone to blame? Well no, I don't think that there is. the QL-had reached the end of its useful life years ago. Long overtaken by faster processors, slicker software and better things. It has clung on to a tenuous existence in these pages and those of Quanta mainly by the good efforts of people like Jochen Merz, Dilwyn Jones and others like them. One of the biggest problems has been that, as the pond shrunk so the fish got a little too big and a mite fractious. In those circumstances the minnows have tended to swim off and hide under the lilly pads - sometimes never to emerge again.

If all this sounds a little like sour grapes because I am now signing off from active service I can assure you that it isn't. There was a policy under most editors of the magazine not to include negative articles and I always felt that was a little wrong. Sometimes a negative viewpoint is what is needed to stimulate activity. It can actually be cathartic and stimulate fresh growth.

Could We have Stopped the Decline?

When the QL first emerged it was as good a contender as any for the crown. All computers in the early 80's were slow, flaky and had clunky software. The QL was as good as any of the others and better than most of them, in fact. There were a few design flaws and a dreadful marketing launch but, on the whole it was a good starting point. The trouble was that was all it was. Once off the mark it behaved just like a British Tennis player. A few good shots then it was straight into the nets. These days nothing can touch a Wintel system - oh yes I can hear all the groans and moans and cries of the LINUX and MAC supporters out there but really, wake up and taste the horlicks. LINUX is an outright failure (I will come to that later) and MACs, while being basically good systems are overpriced and restrictive. Musicians and designers love them but, step outside those niche areas and look at what is available for the MAC. Most street level stuff has to run under emulation.

If you look at it with the rose glasses off the whole computer business is running on empty. Games consoles now account for the bulk of the gaming platforms and the home computer is an email terminal with access to the web for homework, shopping, gambling, holiday and travel booking. The biggest innovation on the home computer in the last few years has been social networking on sites like Facebook and MySpace etc. All the stuff that kept people up all night staring at a screen is now available on something else - even the fleshier pursuits.

I would say, if pushed, that the real decline of our system set in when so many of the productive writers hung up their HEX and stopped innovating. There is no substitute for innovation to keep people interested. As a community we had our share of geek programmers and users. Many of these moved off into LINUX. I know I have said this before but programmers like Jonathan Hudson in many ways fuelled the move away from the QL. If you provide something like QFAX with no proper front end you dangle a carrot before an average user that he cannot hope to cook. Trouble is he has seen the carrot, and, if another provider can do some carrot cake - with icing - he will go there. The same is true of a lot of QL-software. Small, compact and to the point it

may be but Joe public loves a bell and a whistle even if he never blows them.
This is all part of the LINUX argument too.

LINUX

So why do I say that LINUX is a failure? After all there are lots of computers out there running it. Quite a few of the netbook EEPCs run LINUX as a standard O/S and some of that is an active choice by the manufacturers although a lot of the activity is in deciding that adding another £50 onto the cost for a copy of XP Home would push the price tag up too much. The small size of Solid State Memory available also added to this decision since Windoze does tend to take up a lot of space on a drive.

The Netbook and things like it are ideal platforms for LINUX because they come complete. With no expansion slots or unusual hardware to bump up against or conflict with it is a good bet. It is also a platform for net use and is more of a, biggish, smartphone or electronic organiser than a computer so you won't be doing lots of complex stuff on it and won't feel the need for big or sophisticated programs. Ideal for LINUX then.

It is on the desktop that LINUX just does not cut the ketchup - let alone the mustard. Linux itself was only ever going to be the system of choice for two types of user. There is the hardware freak who loves tinkering with bits of electronics and can understand enough code to recompile a driver when one does not exist and there is the basic netbook user described above. Bit of Word Processing, email, net browsing, looking at pictures from your mobile phone. all fine - so long as you have someone who can set the thing up on the PC. UBUNTU is probably the easiest LINUX to set up but even that has its arcane corners in which lurk goblins in bottle glasses asking you obscure questions in unfeasibily squeaky voices.

But compatibility........ Oh dear! Along comes a new printer / graphics card / or other shiny hardware device. Are you really going to wait 6 months for some dusty coder to get one for Christmas and decide to write a driver or interface for it? Of course the LINGANG will chirp up 'write it yourself' but that is not going to happen is it? Your talents probably lie elsewhere and you probably only have LINUX because someone talked you into it and set it up for you. Or because it was free.

There are tools out there to help you. Linspire's Click and Run Warehouse is one but that is the work of the devil to many hardcore LINUX nuts. This is precisely why some QL-programmers moved to LINUX in the first place. A better, faster, hardware platform and people who did not want complete singing and dancing software handed to them on a platter. The problem is that, for the manufacturers of hardware and the mass market software houses, there are not enough LINUX users out there to make it worth adding to the production costs by doing LINUX versions of the software or writing LINUX drivers and there are not enough people using it because there is not enough software or hardware for it. This is the QL problem in Macrocosm. We will never have any new hardware because there is not enough of a user base to pay for its development

Security

Talk to me about virus protection then. OK there are less nasty predators out there for LINUX and the MAC but then there are less of those out there to attack anyway. Of course if you consider that the average MAC user has slung at least four times the amount of money at his setup than the average PC user you might think the online 'phishers of men' would be circling them with currency signs lighting up the eyes. It is not that they are so secure. Every O/S has a swathe of holes in it, unpatched bits that a good programmer can hack into. They all have their vulnerabilities it is just that there are so many PCs running out there that it is easy to break into one. If we all switched to LINUX tomorrow then so would the virus writers and hackers and, oh look, there is all the code on display for them to hack.

Three Dead Trolls In A Baggie summed up the general sentiment in a verse of its snappily named 2001 song Every OS Sucks

"It's free, they say, if you can get it to run The geeks say 'Hey, that's half the fun!' Yeah, well I got a girlfriend and things to get done. The Linux OS sucks!"

Back to the Black Box Then

As I said above a lot of the problems of LINUX are reflected in a small scale on the QL and, even worse, they are compounded by having a growing proportion of people who stick with the older programs when newer, better ones are available. I was sharply reminded of this recently.

One of my older customers called me up and said he had some problems and could he come over to see me. He was using a QL with a Super Gold Card and an EPSON wide carriage dot matrix printer. He used it purely for running ABACUS and the problems were 1. the printer did not print anymore 2. he could not read some of the disks 3. the QL was intermittently giving a corrupt screen

Now I am no hardware repairer but I did have a spare QL here so problem 3 was easy. Problem 2 was also a not a difficult one. He had two sets of floppy drives, one DD and one HD. He was using the DD one and formatting HD disks on it and doing the work on the HD drive which, of course, looked at the disk and said 'the holes say it is HD but the tracks are wrong - must be corrupt' Again easy to solve - tell him not to use the DD drives.

Problem 1 was the stickler. I could test the printer from my laptop using QPC2 (no Windows XP driver for a printer that old). That was OK and I found a small piece of paper jammed into one of the sensors. Cleared that and it worked fine. Luckily the laptop had a parallel port. Many don't these days. Having sorted that I tried it on his system again. Still nothing. I always hated the PSION suite. It was all a bit flaky but, for its time, it was state of the art.

I tried to run the printer edit program but, for the life of me, couldn't recall how to do it. I got fed up with trying to do that on a slow QL and tried to do it on the laptop using QPC2 but it fell over completely. In the end I downloaded a new version of the program and editing programs from Dilwyn's site, edited the BASIC to give me flp1_ and flp2_ instead of mdv1_ and mdv2_ and re-learnt how to do it. It took me all weekend to sort it out but I got it all back up and running.

The crux of this story is this. The user also had a laptop and I suggested getting rid of the QL altogether, buying QPC2 and QPCPrint and running Exchange. As a demo I downloaded the QPC2 demo program, set up a QXL_WIN file with Exchange and some of his data and showed him how to use it.

It was faster than his standard QL. Would print onto his nice laser printer at home (well it would if he bought the full program - I don't think the demo will print) and he could store all the data on his hard drive. He went away to try it out and didn't ever get back in touch. He is probably still using his QL.

If that is the general attitude why do we bother? Great innovation like QDT are not seized upon and the people left seem to exist in a computer museum - frozen in aspic. I reached the age of 60 last week but I still want to take up new stuff and learn to use new equipment and I still want to move forward. I don't read the QL list as much as I used to but when I do I find the same topics rolling round. This week, for example, they were discussing the 36 character file name length. OK it is a tad short but, since no one is doing anything daring in the programming field, what does it matter? I wrote about that in this column way back - at least 10 years ago.

Disagreements and disputes

Over the years there have been a few spats and exchanges in QL Land. Usually someone with strong ideas clashing with a group of people who did not want any charge to the status quo (well Whatever You Want). Some of those spats were downright vicious at times and some appallingly childish but that is what happens in a small community with more than its share of single issue obsessives. Some of these arguments lost us readers and users. There were the obsessivly open source programmers who threw wobblies when their programs were included on cover disks without the reams of source code. And then there was the argument about the SMSQ/E licence. At the original meeting I was of the opinion it should have been free and expressed that. Others disagreed and I went along with their decisions. From my point of view making it free but controlled was the best way to get more people using it and the more people that used it the more software would be written that used all its wonderful features. To be fair to those who wanted to charge for it, the decision was never about the money they could make because we alll knew it would not make any.

There were that failed hardware projects. The Q40 could have been a great QL successor if only its developers had listened to Tony Firshman and Stuart Honeyball (both people who have provided the best hardware ever available for the QL) and treated it as a prototype and made a few, sensible changes to it. Egos, impatience in intractability got in the way. It was still a reasonable machine and a good deal better than a standard QL but not the great thing it could have been.

So I suppose I will bid farewell to all this. As I said at the start I am not sure if I will contribute a column for the next issue - time and motivation are a little short these days. My natural optimism hoped, at the start of QBranch, that the QL would pull something wonderful out of the bag. The potential was always there but it ebbed away. The QBrach email address will still operate so you can get in touch - replies may be slow, It depends on what time I have.

I wish you all a Happy Christmas and and innovative New Year

Goodbye, and thank you for all your support. (and the fish!)

13 Years QL Today ... History and Future

by Jochen Merz

The fact that QBranch is closing the QL business means we need to think about the future of QL Today.

Roy promised to deal with all forthcoming issues of Volume 13, but how are we going to continue after that?

Well, it mainly depends on you, our readers.

First, a bit of history and some more "insights"... I know I have explained some things when renewals were due and we had to raise the price. I do not want to raise the price again, so here are my ideas and considerations.

As most of you know, QL Today was brought to life when IQLR suddenly stopped. Stuart Honeyball and I agreed that we had to have a magazine in order to keep the QL scene together and alive. 13 years ago, the QL scene was quite active ... we had many traders and events ... a bit different from today's situation.

I was quite keen on producing the magazine, and so I did, with the help of many others ... mainly Dilwyn at that time. When we started, we had about 12 pages of advertising, so filling 56 to 60 pages average was much easier. It also meant that the money, which came in from the advertising, could be used to keep the price of QL Today down. Also, 13 years ago, postage, shipping, everything was much cheaper than nowadays. We started with 6 issues per year, which was a lot. Stuart Honeyball mainly shipped the magazines to the very many UK readers those days, so the editorial work, proof reading etc. was shared between Dilwyn Jones and myself (larger share to Dilwyn), and I did all of the layout - of all issues. And I sent out the issues to all the readers who subscribed through me all the time - which is also time consuming, especially for issues with cover disks or CDs (it took days to copy them!)

I also produced German issues of QL today for several years. Also, 6 issues per volume, about 1/3rd of the English volume ... with the help of Dietrich Buder (proof reading), but for a much smaller readership. In the beginning, I also added some sheets for the remainders of the QL Club Germany. Thinking about it, I wonder how I managed to do all this, time-wise.

I don't know if it was a good decision or a bad one, looking at it from today. Most of my QL-ing time was eaten up by QL Today, so, unfortunately, not much time was left for programming and other QL activities. Also, QL Today did not really earn me any money ... so I was forced to do other things to earn money - like the Shareware. Which, again, meant that no time was left for QL programming. And, without updates and no products, no QL income. Which resulted in more pressure to do other things to earn money.

The number of readers decreased, Stuart Honeyball ceased trading, and Roy Wood jumped in. Later, Dilwyn had to stop being QL Today editor for personal reasons and Geoff Wicks became editor. Everything went smoothly, and looking at the years, I think we did very well producing interesting magazines very much in time, well-looking, and very reliable. We lost advertisers, but we did not reduce the size of the magazine - in fact, we filled those pages with editorial content, articles etc. - so in the end the magazine became better value year by year.

I think at the upcoming start of Volumes 8, 9 or 10, when the time pressure was too high, we were suffering from lack of material for the magazine, Roy talked me into continuing at the 6 issue rate per year. Easy for Roy, as it was I who had to spend all the time doing the layouts etc. But, good that he encouraged me to continue. It was clear that 6 issues per year were impossible, so we went to 5, and now for two years, I think, to a quarterly base. Which is much better, as we do not suffer from lack of material as badly as with 6 issues. I do enjoy producing the magazine much more again, especially compared to 3 years ago or so when it was difficult to get enough material together.

The financial situation, subscription price etc. has been explained in previous issues. The situation remains. When we set the subscription price at the middle of a year, we do not know what is going to happen to the postage over the whole of the subscription year. Printing prices have increased, the Dutch Post raises the price every year, mainly at the beginning of January, the UK post raises the prices every now and then and so on.

We have kept the price steady for the last volume, although costs went up ... and you cannot "only" count the costs for producing the magazine and shipping, as I need to care about sending out renewal reminders (a letter to the USA costs EUR 1.70 ... I am not joking!), replacement of lost issues and other things.

With volume 13, the number of readers has fallen under 200. The cost and time per issue goes up with every reader we lose. However, as I will be faced with dealing with all the renewals now (both from Roy and my own ones), I wonder what to do about Volume 14. Do YOU want a volume 14? If so, how do we work out a fair price, not knowing how many readers we will have. To work out a fair price, we have to know how many readers will renew. So, we're stuck.

When we had 500 readers or so, losing 10% caused much less influence to the price of the remaining issues than nowadays. The price for shipping the issues in a big package to the UK (so that Geoff can send them out to the readers, he already offered to do this) will be the same, for 80, 120 or 150 ... but the per-issue price will nearly double between 80 and 150.

As MY subscribers know, I always sent out early renewal reminders, but everybody who paid by credit card was charged after the last magazine of the previous issue was shipped for a while - usually May to July.

Which options do we have? Electronic version of QL Today - not desired by the majority of readers - and speaking to others and knowing for myself - we are flooded with mails and junk mails and ads ... realistically, after 1 or 2 issues the result would not be honoured as much as a printed issue. And you cannot read an electronic issue the same way you can read paper ones, e.g. while taking a bath, while sitting in the sun etc. And, not everybody is on the internet ... and if we go both ways, we just add work and increase the per-issue costs for printed issues considerably! So, no option for me.

I also do not want to raise the price. Everybody has less money to spend, as everything is getting more and more expensive. Also no option for me.

Would it make sense to wait for all renewals to arrive, decide if enough readers exist to justify the work and see, if the magazine can be produced at a specified price, and only accept credit cards and account debit in Germany (so that, if the number is too small, nothing is done and we forget about QL Today?). Mmmhh, no option either, is it?

So, the ONLY way to reduce the costs for the magazine is to reduce the weight of the magazine. The big postage difference comes over 100g. At present, we are under 200g. I checked how many pages including envelope can be used to stay under 100g. The 160g/sqm cover paper could be replaced by 80g. Which would allow 32 pages maximum.

So, opting for 32 pages maximum per issue would reduce postage, printing costs, ensure that we will be able to fill future issues (filling 56 to 60 pages is not always easy, as you know), and would allow me to reduce the subscription price, instead of raising it. In my opinion, the only option. The remaining question is: do we go to DIN A5 size, like Quanta? Actually, we would have more border, footers, headers, and I rather like the A4 size.

As none of us get any younger, and our eyes most likely do not get any better either, not a good option. We could - if you agree - reduce the average character size by 1 point. This should allow me to put over 10% more information on the same number of pages without reducing the text size too much ... going down to A5 gives much, much smaller text, or much less content.

To compensate for the fewer pages even more, we could say that we do not print listings longer than 2 pages anymore, but publish them on the website when the magazine comes out, and add a disk or CD with issue 4 containing all the listings for the readers without internet access. Does that sound like a good and fair option? We would have one heavier issue and more expensive postage for issue 4 only, going this way.

So, there we are. Assuming a moderate price increase and assuming that we do not lose many readers, we could bring the European subscription rate from EUR 33.90 down to something under 30 EUR (and I don't mean 29.99!) and the US rate from 44.90 EUR down by at least 10 EUR or so.

I hope I have explained the reasons and the situation and considerations as well as possible. We (Geoff and I) expect your feedback. We do want to continue QL Today, as you can see and as we feel, it is in important piece to keep the QL scene together as it is, but after all, it is done mainly for YOU.

I cannot really finish this article without thanking Roy for all his encouragement and support, Dilwyn and Geoff, Bruce ... and all our authors and readers - all of you made and make the existence of QL Today possible!

The QL Show Agenda

Quanta Annual General Meeting 2009

QUANTA Committee confirms that the

25th Annual General Meeting

Will be held at the Allesley Hotel, Coventry, England.
CV5 9GP

Members are invited to attend the celebratory workshop opening Saturday April 18th and closing on Sunday April 19th 2009. The AGM will be held at 2.00 pm on Sunday.
Nomination forms for committee members and officers are enclosed with the Quanta magazine.

The programme will include: Talks, Interactive problem solving, Repair desk, Other activities

A celebratory dinner will be held on Saturday night at 7.30 pm
A number of rooms, single, double or twin have been provisionally booked for both Friday and Saturday night. Anyone wishing to come, please contact Quanta secretary Sarah Gilpin at secretary@QUANTA.org.uk
Please state whether you wish to book a room, attend the dinner, provide a talk or demonstration or Set up an activity

The hotel is easily accessible from Birmingham Airport, so we hope our Overseas members will come and join us. Directions will be published in the next Quanta magazine or are available at:
http://www.allesleyhotelcoventry.co.uk/Allesley_Coventry_Hotel_Directions.html

All Quanta workshops were open to members and non-members in the past, so the QL Today team assumes that non-Quanta-members are welcome as well!

MERRY CHRISTMAS AND ALL THE BEST FOR 2009 TO OUR READERS FROM THE QL TODAY TEAM!

The Next Issue

We plan to have the next issue ready for you towards the middle of March 2009. You know: As always, it depends on how quickly we get reviews, articles etc.
You know, we are waiting for YOUR articles and feedback - the magazine lives from it. The more we get and the sooner we get it, the quicker the next issue will be in your hands, and the better it will be. Hope to meet you at one of the forthcoming QL shows - your QL Today Team!